1989-09

Approximate interval estimation methods for the reliability of systems using component data with exponential and Weibull distributions.

Lee, Hyeon-Soo

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

APPROXIMATE INTERVAL ESTIMATION
METHODS FOR THE
RELIABILITYOF SYSTEMS USING COMPONENT
DATA WITH EXPONENTIAL
AND WEIBULL DISTRIBUTIONS

by

LEE, Hyeon-Soo

September, 1989

Thesis Advisor W.M. WOODS

nclassified

urity classification of this page

REPORT DOCUMENTATION PAGE

a Report Security Classification Unclassified		1b Restrictive Markings	
a Security Classification Authority		3 Distribution Availability of Report	
b Declassification Downgrading Schedule		Approved for public release; distribution is unlimited.	
Performing Organization Report Number(s)		5 Monitoring Organization Report Number(s)	
a Name of Performing Organization Naval Postgraduate School	6b Office Symbol *(if applicable)* 55	7a Name of Monitoring Organization Naval Postgraduate School	
c Address *(city, state, and ZIP code)* Monterey, CA 93943-5000		7b Address *(city, state, and ZIP code)* Monterey, CA 93943-5000	
a Name of Funding Sponsoring Organization	8b Office Symbol *(if applicable)*	9 Procurement Instrument Identification Number	
c Address *(city, state, and ZIP code)*		10 Source of Funding Numbers	
		Program Element No / Project No / Task No / Work Unit Accession No	

1 Title *(include security classification)* APPROXIMATE INTERVAL ESTIMATION METHODS FOR THE RELIABILITY OF SYSTEMS USING COMPONENT DATA WITH EXPONENTIAL AND WEIBULL DISTRIBUTIONS

2 Personal Author(s) LEE, Hyeon-Soo

13a Type of Report	13b Time Covered	14 Date of Report *(year, month, day)*	15 Page Count
Master's Thesis	From To	September, 1989	64

6 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms *(continue on reverse if necessary and identify by block number)*
Field	Group	Subgroup	Reliability, Jacknife, Confidence Limit, Exponential,Weibull,Nomal, Newton - Raphson

9 Abstract *(continue on reverse if necessary and identify by block number)*

Two approximate parametric interval estimation methods for system reliability using component test data are developed nd evaluated. One method can be applied to any coherent system with components which have *exponential* failure times ith possibly different failure rates and different mission operating times. This method estimates the ratios of component ilure rates which are then used to develop the approximate lower confidence limit. These ratio estimates are developed with nd without jacknife methods and the two results are compared. This procedure is very accurate and simple to compute, equiring the use of standard chi-square tables. This *ratio* method is subsequently extended to coherent systems with components whose failure times have a *Weibull* distribution. A nearly exact parametric lower confidence limit for $P(X > x)$ is eveloped and evaluated where x is given and X has a *normal* distribution with unknown mean and variance. This procedure also simple to evaluate and requires the use of *Student t* tables.

20 Distribution Availability of Abstract	21 Abstract Security Classification	
☒ unclassified unlimited ☐ same as report ☐ DTIC users	Unclassified	
22a Name of Responsible Individual W.M. WOODS	22b Telephone *(include Area code)* (408) 484-1893	22c Office Symbol 55Wo

D FORM 1473,84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

Approximate Interval Estimation Methods for the Reliability of Systems Using Component Data with Exponential and Weibull Distributions

by

LEE, Hyeon-Soo
Captain, Republic Of Korea Army
B.S., Korea Military Academy, 1982

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September, 1989

ABSTRACT

Two approximate parametric interval estimation methods for system reliability using component test data are developed and evaluated. One method can be applied to any coherent system with components which have *exponential* failure times with possibly different failure rates and different mission operating times. This method estimates the ratios of component failure rates which are then used to develop the approximate lower confidence limit. These ratio estimates are developed with and without jacknife methods and the two results are compared. This procedure is very accurate and simple to compute, requiring the use of standard chi-square tables. This *ratio* method is subsequently extended to coherent systems with components whose failure times have a *Weibull* distribution. A nearly exact parametric lower confidence limit for $P(X > x)$ is developed and evaluated where x is given and X has a *normal* distribution with unknown mean and variance. This procedure is also simple to evaluate and requires the use of *Student t* tables.

TABLE OF CONTENTS

LIST OF TABLES

I. BACKGROUND

Parametric confidence interval procedures for the reliability of mechanical systems are much less developed than procedures for electrical systems. This is due to the more complicated failure distributions used to model mechanical hardware. The failure rate of a *series* system of independent components, with *exponentially* distributed failure times and equal mission times, is the sum of the failure component failure rates. This property has permitted development of numerous methods for system reliability of series and other coherent systems using component failure data. The *Weibull* and *extreme value* distributions have been used in life testing methods for both electrical and mechanical devices. Several methods have been used for obtaining point estimates of parameters for these distributions and for the reliability function itself, Harter and Moore [Ref. 1: pp. 889-901], Mann [Ref. 2: pp. 231-256].

The derivation of simple confidence limits for the reliability function for the *extreme value* distribution with parameters z and δ has posed problems, because methods based on a pivotal quantity such as $(\hat{z} - z)/\hat{\delta}$ are inadequate. Lawless [Ref. 3: pp. 355-364], Johns and Lieberman [Ref. 4: pp. 135-175] and Thoman, Bain and Antle [Ref. 5: pp. 363-372] have developed nearly exact procedures for confidence limits for the reliability function of the *Weibull* and *extreme value* distributions. Schneider and Weissfeld [Ref. 6: pp. 179-186] have developed interval estimation methods for percentiles of the *Weibull* and *extreme value* distribution based on censored data. Although extensive methods have been developed for interval estimates of the reliability of a single component with the *Weibull* or *extreme value* failure distribution, very few parametric interval methods have been developed for system reliability using component test data with *Weibull* failure distributions.

Two approximate parametric interval estimation methods for reliability of coherent systems using component test data are developed and evaluated in this thesis. Evaluations of these procedures are performed using computer simulation for series systems only. One of these methods is developed for the reliability of a series system whose components have *exponential* failure distributions and different mission times. This procedure was found to be quite accurate and can be applied to coherent systems in general. This first procedure is then extended to the case where components of the sys-

tem have *Weibull* failure distributions. The method used is an extension of a non-parametric method developed by Myhre, J., Rosenfeld, A. and Saunders, S. [Ref. 7: pp.213-227].

The *normal* distribution is used extensively in some mechanical reliabilty models. Maximum likelihood and minimum variance unbiased estimators for $P(X \geq x_o)$ when both μ and σ^2 are unknown were developed over thirty years ago, Lieberman and Resnikoff [Ref. 8: pp. 457-516], Folks and others [Ref. 9: pp. 43-50], and Barton [Ref. 10: pp. 227-229]. Exact interval estimation procedures for $P(X \geq x_o)$ were developed by Owen and Hua [Ref. 11: pp.285-311] using the non-central t distribution. Letting X denote strength and Y denote stress, mechanical reliability is sometimes modeled as the value for $P(X > Y)$. Approximate interval estimation procedures for this probability when X and Y are assumed to be normal have been developed by Church and Harris [Ref. 12: pp. 49-54] when the mean and variance of Y are known. Downton [Ref. 13: pp. 551-558] modifies their procedure slightly to get more accurate bounds and suggests an approximate procedure when the means and variances of both X and Y are unknown. Lower confidence intervals for $P(X \geq Y)$ obtained under the assumption of normality for X and Y can lead to serious error when either X or Y or both are truncated well into the tails. Consequently, $P(X \geq x_o)$ may be a more reasonable model of mechanical reliability where x_o is chosen conservatively. A very accurate approximate lower confidence limit procedure for $P(X \geq x_o)$ is developed and evaluated in this thesis. It can be computed easily.

II. INTERVAL ESTIMATION PROCEDURE - EXPONENTIAL CASE

A system of independent components is coherent if an increase in reliability of any one of its components does not cause a degredation in system reliability. Suppose a coherent system has k components. We assume that the failure distribution of component i is *exponential* with failure rate λ_i. Then system reliability R_S can be written as a function of $\lambda_i, t_i, \quad i = 1, 2, \cdots, k; \quad$ *i.e.*,

$$R_S(t) = g(\lambda_1, \lambda_2, \cdots, \lambda_k, t_1, t_2, \cdots, t_k)$$

where $t_i = t_i(t)$ is the operating time for component i, $i = 1, 2, \cdots, k$.

Let $\lambda_m = \max\{\lambda_1, \lambda_2, \cdots, \lambda_k\}$ and $r_i = \lambda_i/\lambda_m \quad i = 1, 2, \cdots, k$. Then one can write

$$R_S(t) = g(\lambda_m, r_1, \cdots, r_k, t_i, \cdots, t_k)$$

If the r_i were known and $\hat{\lambda}_{m,U(\alpha)}$ were an upper $100(1-\alpha)\%$ confidence limit for λ_m, the corresponding lower confidence limit for $R_S(t)$ would be

$$\hat{R}_S(t)_{L(\alpha)} = g(\hat{\lambda}_{m,U(\alpha)}, r_i, \cdots, r_k, t_1, \cdots, t_k)$$

Specifically, if the system is a *series* system of independent components, so that

$$R_S(t) = \exp\left\{-\sum_{i=1}^{k} \lambda_i t_i\right\} = \exp\left\{-\lambda_m \sum_{i=1}^{n} r_i t_i\right\} \tag{2.1}$$

then,

$$\hat{R}_S(t)_{L(\alpha)} = \exp\left\{-\hat{\lambda}_{m,U(\alpha)} \sum_{i=1}^{n} r_i t_i\right\}$$

If n_i items of component i are tested until failure, T_i denotes the total test time accumulated by all n_i items and $n = \sum_{i=1}^{k} n_i$, then $2\lambda_m \sum_{i=1}^{k} r_i T_i$ is χ^2_{2n}. See Bain and Engelhardt [Ref. 14].

An upper confidence limit for λ_m is

$$\hat{\lambda}_{m,U(\alpha)} = \frac{\chi^2_{\alpha,2n}}{2\sum_{i=1}^{k} r_i T_i} \tag{2.2}$$

where $\chi^2_{\alpha,n}$ is the $100(1-\alpha)$th percentile of the χ^2_n distribution. Corresponding equations for truncated testing are similar.

If the r_i are unknown, the following methods estimate the values for r_i from the data. One method uses the likelihood ratio estimate for r_i. The second method uses a jacknifed version of the first method. The two resulting confidence limits $\hat{R}_{S,L(\alpha)}$ and $R_{S,L(\alpha)}$, with and without jacknifing $\hat{r}_i$ respectively, are compared for relative accuracy. Quenouille [Ref. 15: pp. 353-360] first reported a method for estimating ratios that reduced the bias without increasing the variance. Miller [Ref. 16: pp. 1-15] gives an excellent review of the jacknife method which includes a discussion on the application of jacknifing to estimating ratios.

A. LOWER CONFIDENCE LIMIT $R_{S,L}$ WITHOUT JACKNIFING

In this case, the maximum likelihood estimate of the ratio $r_i = \frac{\lambda_i}{\lambda_m}$ is

$$\hat{r}_i = \frac{\hat{\lambda}_i}{\hat{\lambda}_m} \tag{2.3}$$

where $\hat{\lambda}_i = n_i/T_i$ and $\hat{\lambda}_m = \max(\hat{\lambda}_1, \cdots, \hat{\lambda}_k)$. The resulting approximate upper confidence limit for λ_m is

$$\hat{\lambda}_{m,U(\alpha)} = \frac{\chi^2_{\alpha,2n}}{2\sum_{i=1}^{k} \hat{r}_i T_i} \tag{2.4}$$

where $n = \sum_{i=1}^{k} n_i$, and

$$T_i = \sum_{j=1}^{n_i} T_{ij} \qquad i = 1, 2, \cdots, k \tag{2.5}$$

and T_{ij} denotes the failure time of the jth test for component i.

The resulting approximate confidence bound $R_{S,L(\alpha)}$ is given by

$$R_{S,L(\alpha)} = exp\left\{ -\hat{\lambda}_{m,U(\alpha)} \sum_{i=1}^{k} \hat{r}_i t_i \right\} \tag{2.6}$$

B. LOWER CONFIDENCE LIMIT $\hat{R}_{S,L}$ WITH JACKNIFING

The definitions for $\hat{\lambda}_i$, $\hat{r}_i$, n_i , and T_i in Section A are also used in this section. Let $\hat{\lambda}_{ij(\bullet)}$ denote the estimate for λ_i by removing T_{ij} from the data ; *i.e.*,

$$\hat{\lambda}_{ij(\bullet)} = \frac{n_i - 1}{\sum_{l=1}^{n_i} T_{il}} \qquad j = 1, 2, \bullet\bullet\bullet, n_i, \quad l \neq j \tag{2.7}$$

and

$$\hat{\lambda}_{mj(\bullet)} = \frac{n_m - 1}{\sum_{l=1}^{n_m} T_{ml}} \qquad j = 1, 2, \bullet\bullet\bullet, n_m, \quad l \neq j \tag{2.8}$$

Then the jacknifed ratio estimate $\hat{r}_i^*$ is given by

$$\hat{r}_i^* = n^* \hat{r}_i - \frac{(n^* - 1) \sum_{j=1}^{n^*} \hat{r}_{ij(\bullet)}}{n^*} \tag{2.9}$$

where $n^* = \min(n_1, n_2, \bullet\bullet\bullet, n_k)$ and

$$\hat{r}_{ij(\bullet)} = \frac{\hat{\lambda}_{ij(\bullet)}}{\hat{\lambda}_{mj(\bullet)}} \qquad j = 1, 2, \bullet\bullet\bullet, n^* \tag{2.10}$$

Now define $\hat{\lambda}^*_{U(\alpha)}$ by

$$\hat{\lambda}^*_{U(\alpha)} = \frac{\chi^2_{\alpha, 2n}}{2 \sum_{i=1}^{k} \hat{r}_i^* T_i} \quad , \tag{2.11}$$

The corresponding confidence bound $\hat{R}_{S,L(\alpha)}$ is given by

$$\hat{R}_{S,L(\alpha)} = \exp\left\{ -\hat{\lambda}^{*}_{U(\alpha)} \sum_{i=1}^{k} \hat{r}^{*}_{i} t_i \right\} \tag{2.12}$$

When the n_i, $i = 1, 2, \cdots, k$ differ considerably, this jacknife estimation procedure can be unbalanced. That is, the number of data points used to compute the jacknife estimate will differ from one component to another. It was decided to use this jacknifing procedure and determine the effect of differing sample sizes by examining the results of the simulations.

In equation (2.9), the jacknife estimate is constructed by using only the first n^* observations from each component to obtain $\hat{r}_{ij(*)}$ where $n^* = \min\{n_1, n_2, \cdots, n_k\}$. Of course it is rather arbitrary to take the first n^* observations from n_i, since any n^* of the collection of n_i values could be used.

C. SIMULATIONS AND RESULTS FOR EXPONENTIAL CASE

1. Simulation

a. Simulation language and package

The programming language used to simulate this problem was VS FORTRAN on an IBM 3033. Also LEXPN in LLRANII was used to generate observed exponential random variates. SHSORT was used to perform the sort routines.

b. Cases and Input parameters

The six input parameters below determine the conditions for exercising the simulation runs. System reliability is determined by five parameters. Values of these parameters are given in the tables that show the results. The test plan simulated was to test n_i items until all fail.

- number of component types, k : 5 and 15
- system reliability, R_S : 0.9 and 0.975
- significance level, α : 0.05 and 0.2
- component time, t_i : small to large (see tables)
- sample size, n_i : small to large (see tables)

Reliability of a *series* system is expressed as $R_S = \exp\{-\sum_{i=1}^{k} \lambda_i t_i\}$. We chose arbitrarily to determine the failure rates, λ_i, from this equation by assuming all $\lambda_i t_i$ to be equal, consequently,

$$\hat{\lambda}_i = \frac{-\ln R_S}{k\, t_i} \quad .$$

c. Replications

The procedure was replicated 1000 times for each case to get 1000 values of $R_{S,L(\alpha)}$ and 1000 values of $\hat{R}_{S,L(\alpha)}$. We order each set of the 1000 values of $R_{S,L(\alpha)}$ and $\hat{R}_{S,L(\alpha)}$ in ascending order. Then the two approximate confidence bounds, $R_{S,L(\alpha)(1000(1-\alpha))}$ and $\hat{R}_{S,L(\alpha)(1000(1-\alpha))}$, for R_S are the $1000(1-\alpha)$th order values of these two sets of data. Finally, for each of the two ordered sets of data, we find the order indices j_1 and j_2 for which $R_{S,L(\alpha)(j_1)}$ and $\hat{R}_{S,L(\alpha)(j_2)}$ are closest to R_S for their respective sets. Then $j_1 / 1000$ and $j_2 / 1000$ are called the two corresponding simulated true confidence levels.

2. Results

Tables 1 through 3 show the results of the 3 cases simulated. The results indicate that this interval estimation method using estimates of failure rate ratios will yield quite accurate lower confidence limits for system reliability when components have unknown constant failure rates. The jacknife method also yields very accurate confidence limits which are slightly conservative. That is the $100(1-\alpha)$ percentile points of $\hat{R}_{S,L(\alpha)}$, given in the tables, are slightly less than the true value of $\hat{R}_S$. Consquently, $P(\hat{R}_{S,L(\alpha)} \leq R_S) > 1-\alpha$. Alternatively, one can say $\hat{R}_{S,L(\alpha)}$ is a conservative $100(1-\alpha)$ percent lower confidence limit procedure for R_S .

Table 1. RELIABILITY OF A SERIES SYSTEM WITH SMALL NUMBER (LESS THAN 10) OF SAMPLE SIZES - EXPONENTIAL CASE

Number of Compo-nents	Reliability of System	α	Lower Confidence Limit		True Confidence Limit	
			W O Jacknifing	WITH Jacknifing	W O Jacknifing	WITH Jacknifing
5	.90	.2	0.9017	0.8904	0.7720	0.9100
		.05	0.8985	0.8889	0.9620	0.9780
	.975	.2	0.9750	0.9720	0.8020	0.9160
		.05	0.9746	0.9729	0.9600	0.9800
15	.90	.2	0.9019	0.8901	0.7600	0.9500
		.05	0.9012	0.8933	0.9300	0.9800
	.975	.2	0.9745	0.9724	0.8300	0.9600
		.05	0.9746	0.9725	0.9600	0.9800

Sample sizes for 5 components are 9, 7, 10, 8, 6
Sample sizes for 15 components are 6, 7, 5, 6, 7, 8, 9, 5, 8, 7, 9, 10, 7, 9, 6
Component times t_i are 2, 3, 7, 8, 10, 3, 7, 10, 1, 7, 8, 3, 10, 1, 8

Table 2. RELIABILITY OF A SERIES SYSTEM WITH MEDIUM NUMBER (LESS THAN 30) OF SAMPLE SIZES - EXPONENTIAL CASE

Number of Compo-nents	Reliability of System	α	Lower Confidence Limit		True Confidence Limit	
			W O Jacknifing	WITH Jacknifing	W O Jacknifing	WITH Jacknifing
5	.90	.2	0.8999	0.8978	0.8020	0.8660
		.05	0.8990	0.8983	0.9600	0.9680
	.975	.2	0.9750	0.9745	0.8000	0.8460
		.05	0.9748	0.9745	0.9620	0.9740
15	.90	.2	0.8992	0.8960	0.8300	0.9100
		.05	0.8980	0.8981	0.9600	0.9700
	.975	.2	0.9751	0.9738	0.7900	0.9000
		.05	0.9750	0.9741	0.9500	0.9800

Sample sizes for 5 components are 30, 21, 10, 15, 26
Sample sizes for 15 components are 3, 7, 10, 15, 20, 15, 7, 5, 30, 20, 10, 7, 13, 21, 30

Table 3. RELIABILITY OF A SERIES SYSTEM WITH LARGE NUMBER (LESS THAN 100) OF SAMPLE SIZES - EXPONENTIAL CASE

Number of Compo-nents	Reliability of System	α	Lower Confidence Limit		True Confidence Limit	
			W/O Jacknifing	WITH Jacknifing	W/O Jacknifing	WITH Jacknifing
5	.90	.2	0.8999	0.8991	0.8080	0.8200
		.05	0.8990	0.8992	0.9680	0.9620
	.975	.2	0.9750	0.9750	0.8540	0.8060
		.05	0.9751	0.9750	0.9380	0.9520
15	.90	.2	0.9000	0.8988	0.8010	0.8750
		.05	0.9000	0.8996	0.9500	0.9590
	.975	.2	0.9750	0.9747	0.8460	0.8730
		.05	0.9750	0.9750	0.9520	0.9470

Sample sizes for 5 components are 30, 63, 75, 98, 26
Sample sizes for 15 components are 15, 40, 35, 17, 26, 67, 50, 65, 80, 32, 95, 100, 15, 45, 30

III. INTERVAL ESTIMATION PROCEDURE - WEIBULL CASE

A. LOWER CONFIDENCE LIMIT $R_{S,L}$

Consider a *series* system with k components. Let the time to failure X_i of component i have a *Weibull* distribution with density

$$f_i(t_i) = \lambda_i^{\beta_i} \beta\, t_i^{\beta_i - 1} \exp\{-(\lambda_i t_i)^{\beta_i}\} \quad , \qquad t_i > 0 \ . \tag{3.1}$$

Then

$$R_i(t_i) = \exp\{-(\lambda_i t_i)^{\beta_i}\} \quad , \qquad t_i > 0 \tag{3.2}$$

and

$$R_S(t) = \exp\left\{-\sum_{i=1}^{k} \lambda_i^{\beta_i} t_i^{\beta_i}\right\} = \exp\left\{-\lambda_m^{*} \sum_{i=1}^{k} r_i\, t_i^{\beta_i}\right\} \ , \qquad t > 0 \ . \tag{3.3}$$

where $\lambda_i^{*} = \lambda_i^{\beta_i}$, $\lambda_m^{*} = \max_i \lambda_i^{*}$ and $r_i = \lambda_i^{*}/\lambda_m^{*}$. If the β_i are known, $X_i^{\beta_i}$ has constant failure rate $\lambda_i^{\beta_i}$ and the procedures in Chapter II can be used to obtain $\hat{R}_{S,L(\alpha)}$ and $R_{S,L(\alpha)}$ with T_{ij} replaced by $T_{ij}^{\beta_i}$ in equation (2.5). It is well known that $Y \equiv \ln X_i$ has an *extreme value* distribution with CDF

$$F_Y(y) = 1 - \exp\left\{-e^{\frac{y-\xi}{\delta_i}}\right\}$$

where $\xi_i = \ln(1/\lambda_i)$ and $\delta_i = 1/\beta_i$.

Engelhardt and Bain [Ref. 17: p. 323] have developed the following simple unbiased estimators for ξ_i and δ_i , using ordered values $Y_{ij} = \ln X_{i(j)}$

$$\hat{\delta}_i = \frac{1}{\hat{\beta}_i} = \frac{-\sum_{j=1}^{s} Y_{ij} + \frac{s}{n-s} \sum_{j=s+1}^{n} Y_{ij}}{n_i\, k_{n_i}} \tag{3.4}$$

where $s = [0.84 n_i] \equiv$ *largest integer* $\leq 0.84\, n_i$ and $X_{i(j)}$ is the jth order statistics from the sample of size n_i of X_i . Also

$$\hat{\xi}_i = \ln(\frac{1}{\hat{\lambda}}) = \bar{y}_i + \gamma\hat{\delta} \tag{3.5}$$

where $\gamma = 0.5772$ and $\bar{y}_i = \sum_{j=1}^{n_i} Y_{ij}/n_i$. Let

$$\hat{\beta}_i = \frac{1}{\hat{\delta}_i} \tag{3.6}$$

and

$$T_{ij} = X_{ij}^{\hat{\beta}_i} \qquad i = 1, 2, \cdots, n_i \quad j = 1, 2, \cdots, k. \tag{3.7}$$

We approximate the distribution of T_{ij} by the *exponential* distribution with failure rate $\lambda_i^{\hat{\beta}_i} \equiv \lambda_i^*$ and proceed as in Chapter II. Define

$$\lambda_i^* = \frac{n_i}{T_i} \tag{3.8}$$

where $T_i = \sum_{j=1}^{n_i} T_{ij} \quad i = 1, 2, \cdots, k.$ Let $\hat{\lambda}_m^* = \max_i \hat{\lambda}_i^*$ and

$$\hat{r}_i = \frac{\hat{\lambda}_i^*}{\hat{\lambda}_m^*} \tag{3.9}$$

Then an approximate upper confidence limit for $\hat{\lambda}_m^*$ is given by

$$\hat{\lambda}_{m,U(\alpha)}^* = \frac{\chi_{\alpha,2n}^2}{2\sum_{i=1}^{k} \hat{r}_i T_i} \tag{3.10}$$

and the corresponding approximate lower confidence limit $R_{S,L(\alpha)}$ for $R_S(t)$ is given by

$$R_{S,L(\alpha)} = \exp\left\{ -\hat{\lambda}_{m,U(\alpha)}^* \sum_{i=1}^{k} \hat{r}_i t_i^{\hat{\beta}_i} \right\} \tag{3.11}$$

where $n = \sum_{i=1}^{k} n_i$.

This procedure is labeled the Formula procedure in the tables that follow in this section. Its distinguishing feature is the equation for $\hat{\beta}_i$ given by equation (3.6). An alternative procedure, labeled the Newton - Raphson procedure, estimates β_i using the

maximum likelihood procedure which is obtained using a Newton - Raphson approximation method. Equations for $\hat{\beta}_i$ under the Newton - Raphson procedure are provided in Appendix A.

B. SIMULATIONS AND RESULTS

1. Simulation

a. Simulation language and package

The programming language used to simulate this problem was VS FORTRAN on an IBM 3033. Also LEXPN in LLRANII was used to generate observed exponential random variates. SHSORT was used to perform the sort routines.

b. Cases and Input parameters

The six input parameters below determine the conditions for exercising the simulation runs. System reliability is determined by five parameters. Values of these parameters are given in the tables that show the results. The test plan simulated was to test n_i items until all fail.

- number of component types, k : 5 and 15
- system reliability, R_S : 0.9 and 0.975
- significance level, α : 0.05 and 0.2
- component time, t_i : small to large (see tables)
- sample size, n_i : small to large (see tables)

Reliability of a *series* system is expressed as $R_S = \exp\{-\sum_{i=1}^{k} \lambda_i t_i\}$. Failure rates λ_i can be determined from that equation by assuming all $\lambda_i t_i$ to be equal, consequently,

$$\lambda_i = \frac{-\ln R_S}{k\, t_i} \quad .$$

c. Replications

The procedure was replicated 1000 times for each case to get 1000 values of $R_{S,L(\alpha)}$ and 1000 values of $\hat{R}_{S,L(\alpha)}$. We order each set of the 1000 values of $R_{S,L(\alpha)}$ and $\hat{R}_{S,L(\alpha)}$ in ascending order. Then the two approximate confidence bounds, $R_{S,L(\alpha)(1000(1-\alpha))}$ and $\hat{R}_{S,L(\alpha)(1000(1-\alpha))}$, for R_S are the $1000(1-\alpha)$th order values of these two sets of data. Finally, for each of the two ordered sets of data, we find the order indices j_1 and j_2 for which $R_{S,L(\alpha)(j_1)}$ and $\hat{R}_{S,L(\alpha)(j_2)}$ are the closest to R_S for their respective sets. Then $j_1 / 1000$ and $j_2 / 1000$ are called the two corresponding true confidence levels.

2. Results

Tables 4 and 5 display the results of the simulations and determine the accuracy of $\hat{R}_{S,L(\alpha)}$ given in equation (3.11) as a lower confidence limit procedure for system reliability, R_S, for parameter values λ_i, β_i, t_i and R_S given in the tables. The terms in the tables have the same meaning as the corresponding terms in Tables 1 through 3 which were explained in Section 2.C.2 . The procedure would be exact for the Formula method if the values in the Formula column equal the corresponding numbers in the same row in the Reliability of System column. In Table 4 for example, the 80th percentile point of .9205 for the Newton - Raphson procedure is more accurate than the Formula procedure which has an 80th percentile point of .9324. The last column shows, however, that what we have called an 80% lower confidence limit procedure is in fact closer to a 61% procedure. The accuracy improves if the sample size is increased from 15 to 30 as indicated in Table 5. The accuracy improves even more if the number of components in the system increases from 5 to 15 as indicated in Tables 4 and 5.

The results in Tables 4 and 5 show that the $\hat{R}_{L(\alpha)}$ method given by equation (3.11) is too optimistic - especially for small sample sizes and for systems with a small number of components. Modifications to this procedure are needed. These tables also indicate that the Newton - Raphson method is superior to the Formula method.

Table 4. ACCURACY OF $R_{S,L}$ AS A LOWER CONFIDENCE LIMIT FOR R_S WITH SAMPLE SIZES OF 15 - WEIBULL CASE

Number of Components	Reliability of System	α	Lower Confidence Limit		True Confidence Limit	
			Formula	Newton Raphson	Formula	Newton Raphson
5	.90	.2	0.9324	0.9205	0.4820	0.6060
		.05	0.9512	0.9406	0.5380	0.6660
	.975	.2	0.9849	0.9824	0.5040	0.6540
		.05	0.9905	0.9891	0.5600	0.6660
15	.90	.2	0.9193	0.9030	0.5440	0.7620
		.05	0.9416	0.9288	0.6060	0.7760
	.975	.2	0.9802	0.9747	0.6160	0.8080
		.05	0.9865	0.9828	0.6580	0.8240

Initial beta for 5 components are 2.2, 2.4, 2.6, 2.8, 2.5
Initial beta for 15 components are 2.2, 2.4, 2.6, 2.8, 2.5, 2.3, 2.7, 2.5, 2.2, 2.6, 2.9, 2.8, 2.4, 2.6, 2.1
Component times t_i are 2, 3, 7, 8, 10, 3, 7, 10, 1, 7, 8, 3, 10, 1, 8

Table 5. ACCURACY OF $R_{S,L}$ AS A LOWER CONFIDENCE LIMIT FOR R_S WITH SAMPLE SIZES OF 30 - WEIBULL CASE

Number of Components	Reliability of System	α	Lower Confidence Limit		True Confidence Limit	
			Formula	Newton Raphson	Formula	Newton Raphson
5	.90	.2	0.9249	0.9125	0.4100	0.6320
		.05	0.9424	0.9321	0.5100	0.6940
	.975	.2	0.9837	0.9798	0.4380	0.6160
		.05	0.9879	0.9845	0.4600	0.6760
15	.90	.2	0.9207	0.9036	0.5560	0.7200
		.05	0.9333	0.9198	0.4380	0.7520
	.975	.2	0.9808	0.9758	0.5120	0.7520
		.05	0.9850	0.9810	0.5200	0.7920

Initial beta for 5 components are 2.2, 2.4, 2.6, 2.8, 2.5
Initial beta for 15 components are 2.2, 2.4, 2.6, 2.8, 2.5, 2.3, 2.7, 2.5, 2.2, 2.6, 2.9, 2.8, 2.4, 2.6, 2.1

IV. INTERVAL ESTIMATION PROCEDURE - NORMAL CASE

A. BACKGROUND

In recent years it has become popular to model mechanical reliability as $P(X > Y)$ where X denotes strength and Y denotes stress. This formulation of reliability is an average of $P(X > y \mid Y = y)$, since

$$P(X > Y) = E_Y P(X > y) = \int P(X > y) f_y(y)\, dy \quad ,$$

An alternative model is the worst case approach. In the worst case model, one selects a worst case value of y, say y_o , and then designs the strength, X ,of the component so that $P(X > y_o) = R_o$ where R_o is a reliability requirement. If X has a *normal* distribution then this requirement imposes constraints on the mean and variance of X . This is usually done in a manner to comply with standard "safety factor" procedures.

The average model, $P(X > Y)$, uses two random variables and is subject to more random error than the worst case model. The accuracy of the expression $P(X > Y)$ for values of this expression close to one is highly suspect due to deviations in the tail probabilities of both X and Y from those assumed in the model. It is common to assume that both X and Y have normal distributions. *Truncated normal* distributions would be more appropriate for many types of mechanical equipment. Harris and Soms [Ref. 18: pp. 650-663] discuss implications of this problem. Very significant errors in point and interval estimation for reliability are readily demonstrated when X is *truncated normal* but assumed to be *normal* in the more simple model which specifies that $R = P(X > Y)$. Table 10 shows this effect when X is truncated above at $\mu + 1.645\sigma$, where Z_α is the 100 $(1 - \alpha)$th percentile point of the *standard normal* distribution. Church and Harris [Ref. 12: pp. 49-54] and Downton [Ref. 13: pp. 551-558] have developed approximate confidence intervals for $P(X > Y)$ when X is *normal* with unknown mean and variance and Y has the *standard normal* distribution.

B. EXACT AND APPROXIMATE INTERVAL ESTIMATES

Minimum variance unbiased estimators (MVUE) for $R = P(X > y)$ are well known when X is *normal* with unknown mean and known variance and also when the variance is unknown. In the former case, Lieberman and Resnikoff [Ref. 8] developed the result

$$\hat{R} = \Phi\left(\frac{\overline{X} - y}{\sigma\sqrt{\frac{n-1}{n}}} \right)$$

which is MVUE for R where Φ is the *standard normal* cumulative distribution function. When the variance is unknown, several versions of an integral expression for $P(X > y)$ have been developed by Lieberman and Resnikoff [Ref. 8] , Basu [Ref. 19] and Folks and others [Ref. 9].

Exact lower confidence interval estimates for $P(X > y)$ when X is *normal* with unknown mean and variance involve the non-central t - distribution. Owen and Hua [Ref. 11] developed tables of the lower 90% and 95% confidence limit values R_L for $P(X > y)$ based on the non-central t - distribution. These values are tabled for values of k in the range -3.0 (.2) 6.0 and sample sizes n = 2 (1) 18, 21 (3) 30, 40 (20) 100 , where $k = (\overline{x} - y)/s$ and $\overline{x}$ and s are the sample mean and standard deviation. Their tables are essentially exact. An approximation to their exact tabled values is given by R_L^* where $R_L^* \equiv \Phi(\gamma^*)$,

$$\gamma^* = k - \left\{ \frac{1}{n} + \frac{k^2}{2(n - \sqrt{k})} \right\}^{\frac{1}{2}} t_{\alpha, n-1} \quad . \tag{4.1}$$

$k = (\overline{x} - y)/s$ and $t_{\alpha, n-1}$ is the $100(1-\alpha)$th percentile of the t distribution with $n-1$ degrees of freedom. Equation (4.1) was developed in this thesis. It is an extensive ad hoc modification of an equation developed by Church and Harris [Ref. 12]. Tables 6 and 7 display values of $R_L \equiv \Phi(\gamma)$, γ , γ^*, R_L^* , and $R_L^* - R_L$ for k = 1, 2, 3, 4, sample sizes n = 10, 18, 30 and confidence levels 90% and 95% . R_L and γ denote the "exact" lower confidence limits and corresponding $\Phi^{-1}(R_L)$ values from Owen and Hua [Ref. 11]. Both γ^* and R_L^* are given by equation (4.1). The accuracy of the approximate confidence interval is quite good relative to the values for R_L given by Owen and Hua [Ref. 11].

Table 6. APPROXIMATE (R_L^*) AND EXACT (R_L) 90% CONFIDENCE LIMITS FOR $P(X > Y_o)$

n	$t_{.10,n-1}$	k	R_L:$\Phi(\gamma)$	γ	γ^*	R_L^*:$\Phi(\gamma^*)$	$R_L^* - R_L$
10	1.3830	1	0.68156	0.47194	0.45454	0.67528	-0.00628
		2	0.89130	1.23397	1.20199	0.88517	-0.00613
		3	0.97453	1.95262	1.88991	0.97025	-0.00428
		4	0.99602	2.65302	2.54950	0.99437	-0.00165
18	1.3334	1	0.73037	0.61385	0.61133	0.72950	-0.00087
		2	0.92569	1.44512	1.44038	0.92488	-0.00081
		3	0.98755	2.24314	2.23150	0.98684	-0.00071
		4	0.99877	3.02679	3.00614	0.99859	-0.00018
30	1.3114	1	0.75937	0.70424	0.70508	0.75960	0.00023
		2	0.94256	1.57740	1.57852	0.94248	-0.00008
		3	0.99233	2.42404	2.42459	0.99205	-0.00028
		4	0.99945	3.26207	3.25926	0.99940	-0.00005

Table 7. APPROXIMATE (R_L^*) AND EXACT (R_L) 95% CONFIDENCE LIMITS FOR $P(X > Y_o)$

n	$t_{.05,n-1}$	k	R_L:$\Phi(\gamma)$	γ	γ^*	R_L^*:$\Phi(\gamma^*)$	$R_L^* - R_L$
10	1.8331	1	0.63052	0.33311	0.27702	0.60912	-0.02140
		2	0.85187	1.04477	0.94228	0.82692	-0.02495
		3	0.95565	1.70308	1.52864	0.93655	-0.01910
		4	0.99031	2.33813	2.07743	0.98075	-0.00956
18	1.7396	1	0.69504	0.51007	0.49292	0.68896	-0.00608
		2	0.90348	1.30221	1.26991	0.89777	-0.00571
		3	0.97996	2.05344	1.99739	0.97674	-0.00322
		4	0.99735	2.78717	2.70338	0.99638	-0.00097
30	1.6991	1	0.73361	0.62369	0.61789	0.73167	-0.00194
		2	0.92855	1.46579	1.45391	0.92677	-0.00178
		3	0.98855	2.27522	2.25448	0.98758	-0.00097
		4	0.99894	3.07140	3.04028	0.99873	-0.00021

Tables 8 and 9 display the results of computer simulations with 1,000 replications to check the accuracy of the R_L^* method for 80% and 90% lower confidence limits for $P(X > y) \equiv R$ for $y = 3$, with various values of σ and μ determined so that R equals the values shown. The procedure would be exact at the 80% level if the values in the column labelled $\alpha = .2$ equal the corresponding values of R in the same row. The "true" confidence level corresponds to the index $i(R)$ of 1,000 ordered values of $R^*_{L(\alpha)}$ for which $R^*_{L(\alpha),i(R)} = R$. For example, in the seventh row of Table 8, R = .950, $\sigma_x = 20$, N = 10, $R^*_{L(.20),800}$ was .9575, and $R^*_{L(.10),900}$ was .9533. Also $R^*_{L(.20),758} = .950$ and $R^*_{L(.10),886} = .950$. Tables 8 and 9 indicate that the R_L^* procedure given by equation (4.1) is quite accurate at the 80% level of confidence for the cases simulated.

Table 8. ACCURACY ANALYSIS OF R_L^* PROCEDURE FOR 80% AND 90% LOWER CONFIDENCE LIMITS FOR $P(X > 3)$ WHEN X IS NORMALLY DISTRIBUTED

R	σ_x	n	$R_{L,100(1-\alpha)}$		True Confidence Level	
			$\alpha = .2$	$\alpha = .1$	$\alpha = .2$	$\alpha = .1$
.950	0.5	10	0.9624	0.9504	0.7310	0.8970
		25	0.9546	0.9542	0.7560	0.8820
		75	0.9517	0.9501	0.7800	0.8980
	1.0	10	0.9606	0.9531	0.7470	0.8910
		25	0.9501	0.9552	0.7970	0.8770
		75	0.9509	0.9528	0.7900	0.8770
	20.0	10	0.9575	0.9533	0.7580	0.8860
		25	0.9558	0.9551	0.7410	0.8730
		75	0.9511	0.9511	0.7830	0.8820
.990	0.5	10	0.9924	0.9913	1.0000	0.8840
		25	0.9906	0.9898	0.7800	0.9070
		75	0.9903	0.9907	0.9800	0.8780
	1.0	10	0.9921	0.9899	0.9990	0.9000
		25	0.9913	0.9904	0.7570	0.8910
		75	0.9904	0.9898	0.7800	0.9060
	20.0	10	0.9923	0.9922	0.9880	0.8750
		25	0.9909	0.9908	0.7760	0.8800
		75	0.9902	0.9903	0.7940	0.8920
.995	0.5	10	0.9966	0.9948	1.0000	0.9030
		25	0.9955	0.9949	0.7790	0.9020
		75	0.9956	0.9949	0.7540	0.9060
	1.0	10	0.9966	0.9952	1.0000	0.8950
		25	0.9960	0.9955	0.8630	0.8880
		75	0.9949	0.9947	0.8040	0.9150
	20.0	10	0.9970	0.9951	1.0000	1.0000
		25	0.9960	0.9957	0.7720	0.8700
		75	0.9951	0.9948	0.7940	0.9140

Table 9. ACCURACY ANALYSIS OF R_L^* PROCEDURE FOR 80% AND 90% LOWER CONFIDENCE LIMITS FOR $P(X > 30)$ WHEN X IS NORMALLY DISTRIBUTED

R	σ_x	n	$R_{L,1000(1-\alpha)}$		True Confidence Level	
			α = .2	α = .1	α = .2	α = .1
.950	0.5	10	0.9621	0.9466	0.7510	0.9000
		25	0.9530	0.9519	0.7680	0.8890
		75	0.9507	0.9491	0.7880	0.9070
	1.0	10	0.9605	0.9528	0.7480	0.8910
		25	0.9497	0.9549	0.8030	0.8810
		75	0.9509	0.9527	0.7910	0.8780
	20.0	10	0.9575	0.9533	0.7580	0.8860
		25	0.9558	0.9551	0.7410	0.8730
		75	0.9511	0.9511	0.7830	0.8820
.990	0.5	10	0.9923	0.9911	1.0000	0.8860
		25	0.9900	0.9890	0.7990	0.9140
		75	0.9901	0.9904	0.7980	0.8860
	1.0	10	0.9921	0.9899	0.9990	0.9310
		25	0.9911	0.9902	0.7620	0.8950
		75	0.9904	0.9898	0.7800	0.9070
	20.0	10	0.9923	0.9922	0.9880	0.8750
		25	0.9909	0.9908	0.7760	0.8800
		75	0.9902	0.9903	0.7940	0.8840
.995	0.5	10	0.9965	0.9946	1.0000	0.9040
		25	0.9952	0.9946	0.7900	0.9140
		75	0.9955	0.9948	0.7620	0.9100
	1.0	10	0.9966	0.9951	1.0000	0.9280
		25	0.9959	0.9953	0.7610	0.8920
		75	0.9949	0.9947	0.8040	0.9140
	20.0	10	0.9970	0.9951	1.0000	1.0000
		25	0.9960	0.9957	0.7600	0.8700
		75	0.9951	0.9948	0.7940	0.9340

Table 10 displays the inaccuracies of the R_L^* lower confidence limit procedure for $P(X > 3)$ using equation (4.1), which assumes X is $N(\mu, \sigma^2)$, when in fact X has the distribution of a *normal* random variable with mean μ and variance σ^2 that has been truncated at $\mu + 1.645\sigma$. Note that $\mu + 1.645\sigma >> 3$, because $P(X > 3) = .95, .99$ and $.995$.

Examination of Table 10 reveals gross inaccuracies; consequently, even when the distribution of X is truncated far above the value y, the exact lower confidence limit for $P(X > y)$ can be greatly in error when computed under the assumption that X is *normal*. This problem will be compounded when one is computing "exact" confidence intervals for $P(X > Y)$ assuming both X and Y are *normal* when in fact one or both may be *truncated normal*. This suggests that modeling mechanical reliability as $P(X > Y)$ may be more risky than more standard engineering approachs for modeling mechanical reliability which include the notion of safety factors.

Table 10. ACCURACY ANALYSIS OF R_L^* PROCEDURE FOR 80% AND 90% LOWER CONFIDENCE LIMITS FOR $P(X > 3)$ WHEN X IS TRUNCATED NORMAL

Reliability of System	σ_x	n	$R_{L,1000(1-\alpha)}$		True Confidence Level	
			$\alpha = .2$	$\alpha = .1$	$\alpha = .2$	$\alpha = .1$
.950	0.5	10	0.9806	0.0566	1.0000	1.0000
		25	0.8941	0.0125	1.0000	1.0000
		75	0.9383	0.0871	1.0000	1.0000
	1.0	10	0.8904	0.0419	1.0000	1.0000
		25	0.9555	0.0813	1.0000	1.0000
		75	0.9431	0.0782	1.0000	1.0000
	20.0	10	0.8503	0.0105	1.0000	1.0000
		25	0.9274	0.0485	1.0000	1.0000
		75	0.9496	0.1222	1.0000	1.0000
.990	0.5	10	0.9734	0.6035	1.0000	1.0000
		25	0.9924	0.6346	1.0000	1.0000
		75	0.9780	0.6552	1.0000	1.0000
	1.0	10	0.9976	0.9172	1.0000	1.0000
		25	0.9884	0.7505	1.0000	1.0000
		75	0.9951	0.6800	1.0000	1.0000
	20.0	10	0.9593	0.5116	1.0000	1.0000
		25	0.9989	0.6821	1.0000	1.0000
		75	0.9883	0.7514	1.0000	1.0000
.995	0.5	10	0.9813	0.7328	1.0000	1.0000
		25	0.9719	0.9096	1.0000	1.0000
		75	0.9933	0.9130	1.0000	1.0000
	1.0	10	0.9709	0.9015	1.0000	1.0000
		25	0.9991	0.8996	1.0000	1.0000
		75	0.9949	0.9267	1.0000	1.0000
	20.0	10	0.9723	0.6432	1.0000	1.0000
		25	0.9923	0.7539	1.0000	1.0000
		75	0.9971	0.9489	1.0000	1.0000

V. CONCLUSIONS AND RECOMMENDATIONS

The lower interval estimation procedure for reliability of coherent systems which was developed in Chapter II appears to be accurate, easy to use, and applicable to coherent systems. Although this procedure assumes that failure times of all components of the system are independent and have *exponential* probability distributions, it can be easily extended to systems with component failure distributions that can be transformed into *exponential* failure distributions ; e.g., *Weibull* distribution with known shape parameter. This procedure has potential for being combined with a similar procedure for systems with cyclical components. The combined procedure would provide for use of *binomial* component test data and *exponential* component test data to compute lower confidence limits on the reliability of coherent systems with both cyclic and continuous components. Such an extension could use a failure rate ratio estimation procedure similar to that developed in this thesis. Such a method should be explored.

The interval estimation method for the reliability of a system with components that have *Weibull* failure times is not sufficiently accurate to be applied to systems that have 10 or fewer components each with ten or fewer tests. This procedure needs further study and refinement.

The approximate lower confidence limit for component reliability $P(X > y)$ when X is *normally* distributed with unknown mean and variance is very accurate. It does not require an extensive set of tables such as those developed by Owen and Hua[Ref. 11], but only requires the use of the standard t tables.

APPENDIX A. MLE OF WEIBULL PARAMETERS BY THE NEWTON - RAPHSON METHOD

Let $X \sim WEI(\alpha, \beta)$. Then the likelihood function for the first r ordered observations from a random sample of size n is given by

$$f(x_{1:n}, \cdots, x_{r:n}) = \frac{n!}{(n-r)!}\left[\prod_{i=1}^{r} f_X(x_{i:n})\right][1 - F_X(x_{r:n})]^{n-r}$$

$$= \frac{n!}{(n-r)!}\left(\frac{\beta}{\alpha}\right)^r \prod_{i=1}^{r}\left(\frac{x_{i:n}}{\alpha}\right)^{\beta-1} \exp\left\{-\left[\sum_{i=1}^{r}\left(\frac{x_{i:n}}{\alpha}\right)^{\beta} + (n-r)\left(\frac{x_{r:n}}{\alpha}\right)^{\beta}\right]\right\}$$

Setting the partial derivatives of this likelihood with respect to α and β equal to zero gives the MLE's $\hat{\alpha}$ and $\hat{\beta}$ as solutions to the equations

$$\frac{\sum_{i=1}^{n} x_i^{\hat{\beta}} \ln x_i}{\sum_{i=1}^{n} x_i^{\hat{\beta}}} - \frac{1}{\beta} = \frac{1}{n}\sum_{i=1}^{n} \ln x_i$$

and

$$\hat{\alpha} = \left[\frac{\sum_{i=1}^{n} x_i^{\hat{\beta}}}{n}\right]^{\frac{1}{\hat{\beta}}}$$

where $n = r$. It can be shown that these equations have unique solutions which are the maximum likelihood estimates. The NEWTON - RAPHSON method for solving an equation $g(\hat{\beta}) = 0$ is to determine successive approximations $\hat{\beta}_j$, where $\hat{\beta}_{j+1} = \hat{\beta}_j - g(\hat{\beta}_j)/g'(\hat{\beta}_j)$. Therefore, the estimates of α and β can be solved by letting

$$g(\hat{\beta}) = \frac{\sum_{i=1}^{n} x_i^{\hat{\beta}} \ln x_i}{\sum_{i=1}^{n} x_i^{\hat{\beta}}} - \frac{1}{\beta} - \frac{1}{n}\sum_{i=1}^{n} \ln x_i \quad .$$

The derivative of $g(\hat{\beta})$ is

$$g'(\hat{\beta}) = \frac{\sum_{i=1}^{n} x_i^{\hat{\beta}} (\ln x_i)^2}{\sum_{i=1}^{n} x_i^{\hat{\beta}}} - \frac{\sum_{i=1}^{n} x_i^{\hat{\beta}} \ln x_i}{\sum_{i=1}^{n} x_i^{\hat{\beta}}} + \left(\frac{1}{\hat{\beta}}\right)^2 \quad .$$

APPENDIX B. FORTRAN CODE FOR INTERVAL ESTIMATION PROCEDURE - EXPONENTIAL CASE

```
      PROGRAM EXPONE
C
C     ***************************************************************************
C     *                                                                         *
C     *   THIS IS A PROGRAM TO COMPUTE THE TRUE CONFIDENCE LIMIT OF             *
C     *   A SERIES SYSTEM  WITH / WITHOUT  JACKNIFE METHOD  AND                 *
C     *   COMPARE THE DIFFERENCE OF THOSE RESULTS.                              *
C     *                                                                         *
C     *   BELOWS ARE GIVEN OR ASSUMED.                                          *
C     *                                                                         *
C     *        N      ; NUMBER OF COMPONENTS TYPE                             *
C     *        RSYS   ; RELIABILITY OF A SERIES SYSTEM                        *
C     *        ALPHA  ; SIGNIFICANCE LEVEL                                    *
C     *        TIME   ; TEST TIME                                             *
C     *        SAMPLE ; SAMPLE SIZE FOR EACH TYPE OF COMPONENT                *
C     *                                                                         *
C     *                                                                         *
C     *   THESE ARE VARIABLES USED.                                             *
C     *                                                                         *
C     *        B      ; TEMPORARY ARRAY FOR EXPONENTIAL RANDOM VARIATE        *
C     *        BIGLAM ; LARGEST VALUE OF LAMBDA                               *
C     *        CASE   ; COUNTER                                               *
C     *        CHISQR ; CHI-SQUARE VALUE FOR GIVEN ALPHA AND DF               *
C     *        CLR1   ; TRUE CONFIDENCE LIMIT FROM NON-JACKNIFING             *
C     *        CLR2   ; TRUE CONFIDENCE LIMIT FROM JACKNIFING                 *
C     *        CTIME  ; TEST TIME OF INDIVIDUAL COMPONENT                     *
C     *        DF     ; DEGREE OF FREEDOM FOR CHI-SQUARE.                     *
C     *        DIFF   ; ABSOLUTE VALUE OF DIFFERENCE BETWEEN R1 & R2          *
C     *        LAMBDA ; FAILURE RATE OF EACH COMPONENT TYPE                   *
C     *        LAMHAT ; LAMBDA HAT FOR JACKNIFING                             *
C     *        LAMHM  ; FINAL LAMBDA HAT FROM NON-JACKNIFING                  *
C     *        LAMHST ; FINAL LAMBDA HAT FROM JACKNIFING                      *
C     *        LAMMAX ; LARGEST VALUE OF LAMHAT                               *
C     *        LOMIT  ; LAMBDA WITH 1 COMPONENT OMITTED                       *
C     *        KEY1   ; TEMPORARY ARRAY                                       *
C     *        KEY2   ; TEMPORARY ARRAY                                       *
C     *        R      ; INITIALLY COMPUTED RELIABILITY FOR EACH               *
C     *                 COMPONENT TYPE                                        *
C     *        R1     ; COMPUTED RELIABILITY BY NON-JACKNIFING                *
C     *        R2     ; COMPUTED RELIABILITY BY JACKNIFING                    *
C     *        RATIO1 ; RATIO OF LAMBDA FOR NON-JACKNIFING                    *
C     *        RATIO2 ; RATIO OF LAMBDA FOR JACKNIFING                        *
C     *        ROMIT  ; RATIO  WITH 1 COMPONENT OMITTED                       *
C     *        ROMSUM ; USED FOR JACKNIFE, SUM OF R WITH OMIT 1               *
C     *        RSTAR  ; FINALLY COMPUTED R VALUE ( R STAR )                   *
C     *        RVAL1  ; R( 500 * (1-ALPHA) ) FOR NON-JACKNIFING               *
C     *        RVAL2  ; R( 500 * (1-ALPHA) ) FOR JACKNIFING                   *
C     *        T      ; TOTAL TEST TIME OF EACH COMPONENT TYPE                *
C     *        ZALPHA ; RIGHT PERCENTILE POINT(NORMAL DISTRIBUTION)           *
```

```
C     *                                                                      *
C     ************************************************************************
C
C
      PARAMETER (NN = 500)

      REAL     TIME(15), ALPHA(2), ZALPHA(2), RSYS(2), LAMBDA(15), P
      REAL     LAMMAX, LAMHAT(15), B(100), RSTAR(15), LAMHM
      REAL     RATIO1(15), RATIO2(15), BIGLAM, DIFF(15)
      REAL     LOMIT(15,100), CTIME(15,100), ROMIT(15,100), T(15)
      REAL     ROMSUM , SUM1, SUM2, LAMHST, R1(NN), R2(NN), KEY1(NN)
      REAL     RVAL1(15), RVAL2(15), CLR1(15), CLR2(15), KEY2(NN)
      REAL     DIFFR1(15), DIFFR2(15), R(15)

      INTEGER  SAMPLE(15), N(2), DF, CASE

      DATA  N      /  5,0  /
      DATA  RSYS   / .9, .975 /
      DATA  TIME   / 2,3,7,8,10,3,7,10,1,7,8,3,10,1,8 /
      DATA  SAMPLE / 30,63,75,98,26,15,7,5,30,20,10,7,13,21,30 /
      DATA  ALPHA  / .2, .05 /
      DATA  ISEED  / 1736 /


C     /*  SUBROUTINE ZTABLE COMPUTES THE RIGHT PERCENT POINT ZALPHA   */
C     /*  FROM RIGHT CUMULATIVE PROBABILITY ALPHA                     */

      DO 10 I = 1, 2
           CALL ZTABLE(ALPHA(I), ZALPHA(I))
   10 CONTINUE


      CASE = 0

      /* II IS INDEX FOR N                                            */

      DO 150 II = 1, 2

C          /* COMPUTE THE DEGREE OF FREEDOM FOR CHI-SQUARE            */

           DF = 0.
           DO 20 I = 1, N(II)
                DF = DF + SAMPLE(I)
   20      CONTINUE
           DF = 2 * DF


C          /* FINDING COMPONENT TYPE THAT HAS THE MINIMUM NUMBER      */
C          /* OF SAMPLE SIZE                                          */

           NSTAR = 999
           DO 25 I = 1, N(II)
                IF (SAMPLE(I) .LE. NSTAR) THEN
                     NSTAR = SAMPLE(I)
                ENDIF
   25      CONTINUE
```

```
            /* JJ IS INDEX FOR RSYS                                          */
            DO 140 JJ = 1, 2

C                 /* COMPUTE LAMBDA FROM THE GIVEN EQUATION  AND             */
C                 /* FIND THE BIGGEST LAMBDA                                 */

                  BIGLAM  = 0
                  DO 30 K = 1, N(II)
                       LAMBDA(K) = ( -ALOG(RSYS(JJ)) / N(II)) / TIME(K)
                       IF( LAMBDA(K) .GE. BIGLAM)  BIGLAM = LAMBDA(K)
                       R(K) = EXP( - LAMBDA(K) * TIME(K) )
   30              CONTINUE


C                 /* COMPUTE CHI-SQUARE VALUE FOR GIVEN ALPHA AND DF         */


                  /* LL IS INDEX FOR ALPHA                                   */
                  DO 130 LL = 1, 2

                       IF ( DF .EQ. 1 ) THEN
                            P = ALPHA(LL) / 2
                            CALL ZTABLE(P, ZALPHA(LL))
                            CHISQR = ZALPHA(LL) ** 2

                       ELSE IF ( DF .EQ. 2 ) THEN
                            CHISQR = -2 * ALOG(ALPHA(LL))

                       ELSE IF ( DF .GE. 3 ) THEN
                            CHISQR = DF * (1 - 2./(9 * DF) +
     *                                ZALPHA(LL) * SQRT( 2./(9 * DF))) ** 3
                       ENDIF



                       CASE = CASE + 1
                       DIFF(CASE) = 0

                       DO 120 L = 1, NN

                            LAMMAX = -99.
                            DO 50 I = 1, N(II)


C                                /* GENERATE EXPONENTIAL RANDOM             */
C                                /* NUMBERS WITH MU = 1                     */

                                 CALL LEXPN(ISEED, B, SAMPLE(I), 1, 0)
                                 T(I) = 0

                                 DO 40 J = 1, SAMPLE(I)

C                                     /* CONVERT TO EXPONENTIAL RANDOM      */
C                                     /* NUMBERS WITH MU = LAMBDA AND       */
```

```
C                                   /* ADD THOSE FOR EACH COMPONENT   */
C                                   /* TYPE                           */

                                    B(J) = B(J) / LAMBDA(I)
                                    T(I) = T(I) + B(J)
                                    CTIME(I,J) = B(J)
   40                          CONTINUE

                               LAMHAT(I) = SAMPLE(I) / T(I)


C                              /* FINDING MAXIMUM LAMBDA HAT AND ITS  */
C                              /* INDEX                               */

                               IF ( LAMHAT(I) .GE. LAMMAX ) THEN
                                    M = I
                                    LAMMAX = LAMHAT(I)
                               ENDIF

   50                     CONTINUE


C                         /* RATIO1 IS FOR WITHOUT JACKNIFE           */
C                         /* RATIO2 IS FOR WITH    JACKNIFE           */

                          DO 60 I = 1, N(II)
                               RATIO1(I) = LAMBDA(I) / BIGLAM
                               RATIO2(I) = LAMHAT(I) / LAMMAX
   60                     CONTINUE



C                         /* PART OF JACKNIFE METHOD FOR LAMBDA       */
C                         /* WITH OMIT 1 VARIABLE EACH TIME           */

                          DO 90 I = 1, N(II)
                             DO 80 J = 1, SAMPLE(I)

                                LOMIT(I,J) = 0.
                                DO 70 K = 1, SAMPLE(I)

                                  IF (J .NE. K) THEN
                                    LOMIT(I,J) = LOMIT(I,J) + CTIME(I,K)
                                  ENDIF

   70                             CONTINUE

                                  LOMIT(I,J) = (SAMPLE(I)-1) / LOMIT(I,J)

   80                        CONTINUE
   90                     CONTINUE
```

```
C                       /* ADAPT ABOVE RESULT TO OUR EQUATION TO    */
C                       /* GET THE RELIABILITY AND TRUE CONFIDENCE  */
C                       /* LIMIT                                    */

                        SUM1 = 0.
                        SUM2 = 0.
                        SUM3 = 0.
                        SUM4 = 0.

                        DO 110 I = 1, N(II)


C                          /* NON JACKNIFING ( ORIGINAL )           */

                           SUM1 = SUM1 + RATIO1(I) * T(I)
                           SUM2 = SUM2 + RATIO1(I) * TIME(I)


C                          /* WITH JACKNIFING                       */

                           ROMSUM = 0.
                           DO 100 J = 1, NSTAR
                              ROMIT(I,J) = LOMIT(I,J) / LOMIT(M,J)
                              ROMSUM     = ROMSUM + ROMIT(I,J)
  100                      CONTINUE

                           RSTAR(I) = NSTAR * RATIO2(I) -
     *                               (NSTAR - 1) * ROMSUM / NSTAR
                           SUM3 = SUM3 + RSTAR(I) * T(I)
                           SUM4 = SUM4 + RSTAR(I) * TIME(I)
  110                   CONTINUE



C                       /* R1 ; RELIABILITY OF A SYSTEM WITHOUT     */
C                       /*      JACKNIFE                            */
C                       /* R2 ; RELIABILITY OF A SYSTEM WITH        */
C                       /*      JACKNIFE                            */

                        LAMHM = CHISQR / (2 * SUM1)
                        R1(L) = EXP( - LAMHM  * SUM2 )

                        LAMHST  = CHISQR / (2 * SUM3)
                        R2(L)   = EXP( - LAMHST * SUM4 )


                        IF ( ABS(R1(L) - R2(L)) .GE. DIFF(CASE) ) THEN
                             DIFF(CASE) = ABS(R1(L) - R2(L))
                             DIFFR1(CASE) = R1(L)
                             DIFFR2(CASE) = R2(L)
                        ENDIF

  120                CONTINUE

C                    /* NONIMSL LIBRARY 'SHSORT' WILL SORT R1, R2  */
C                    /* BY SHELL SORT ALGORITHM                    */
```

```
                    CALL SHSORT(R1, KEY1, NN)
                    CALL SHSORT(R2, KEY2, NN)

                    MM = NN * (1 - ALPHA(LL))

                    RVAL1(CASE) = R1(MM)
                    RVAL2(CASE) = R2(MM)


C                   /* SUBROUTINE FINDJ FINDS THE INDEX OF R1, R2    */
C                   /* WHICH THE VALUE OF IT IS CLOSEST TO RSYS.     */

                    CALL FINDJ(R1, NN, RSYS(JJ), J1)
                    CALL FINDJ(R2, NN, RSYS(JJ), J2)

                    CLR1(CASE) = J1 / FLOAT(NN)
                    CLR2(CASE) = J2 / FLOAT(NN)

  130           CONTINUE
  140       CONTINUE
  150 CONTINUE



      WRITE(6,600)
      WRITE(6,650)
      WRITE(6,670)

      CASE = 1
      DO 210 II = 1, 2
         DO 200 JJ = 1, 2
            DO 190 LL = 1, 2

               WRITE(6,700) N(II), RSYS(JJ), ALPHA(LL), RVAL1(CASE),
     *              RVAL2(CASE), DIFF(CASE), CLR1(CASE), CLR2(CASE)

               WRITE(6,888) DIFFR1(CASE), DIFFR2(CASE)
               CASE = CASE + 1

  190       CONTINUE
  200    CONTINUE
         WRITE(6,777) ( SAMPLE(J), J=1,N(II) )
         WRITE(6,999) ( R(J), J=1,N(II) )
  210 CONTINUE



  600 FORMAT('1',5(/),7X,'**** RELIABILITY OF SERIES SYSTEM ****')
  650 FORMAT(///,T50,'R1, CL1 ; WITHOUT JACKNIFING',
     *         /,T50,'R2, CL2 ; WITH    JACKNIFING')
  670 FORMAT(///,T6,'NUMBER OF',T19,'RELIABILITY',T33,'ALPHA',T46,'R1',
     *       T56,'R2',T63,'MAX (R1 - R2)',T79,'TRUE',T87,'TRUE',/,
     *       T6,'COMPONENTS',T19,'OF SYSTEM',T79,'C.L. 1',T87,'C.L. 2',
     *       /,T5,89('-'))
```

```
  700 FORMAT(/,T8,I5,T22,F5.3,T33,F5.2,T43,F7.4,T53,F7.4,T65,F8.4,T78,
     *       F7.4,T86,F7.4)

  777  FORMAT(/,T3,'SAMPLE SIZE ARE ', 5(2X,I3) )
  888  FORMAT(T60,'R1=',F6.4,'  R2=',F6.4)
  999  FORMAT(/,T3,'R(I) ; ',5(1X,F5.3))

       STOP
       END
```

```
*******************************************************************************
*                                                                             *
*******************************************************************************

       SUBROUTINE ZTABLE(ALPHA,ZALPHA)
C
C      << SUBROUTINE ZTABL2 COMPUTES RIGHT PERCENT POINT ZALPHA FROM  >>
C      << RIGHT CUMULATIVE PROBABILITY ALPHA                          >>
C
       REAL ALPHA, ZALPHA

       IF ((ALPHA .GT. 0.0) .OR. (ALPHA .LT. 1)) THEN

            W = - ALOG(4 * ALPHA * ( 1 - ALPHA ))
            ZALPHA = SQRT(W * (2.06118 - ( 5.72622 / (W + 11.6406))))

            IF (ALPHA .GT. 0.5) THEN
                 ZALPHA = - ZALPHA
            ENDIF
       ENDIF

       RETURN
       END
```

```
*******************************************************************************
*                                                                             *
*******************************************************************************

       SUBROUTINE SHSORT(A,KEY,N)
       DIMENSION A(N),KEY(N)
       M1=1
     6 M1=M1*2
       IF (M1 .LE. N)  GO TO 6
       M1=M1/2-1
       MM=MAX0(M1/2,1)
       GO TO 21
    20 MM=MM/2
       IF (MM .LE. 0)  GO TO 100
    21 K=N-MM
```

```
   22 DO 1 J=1,K
      II=J
   11 IM=II+MM
      IF (A(IM) .GE. A(II))  GO TO 1
      TEMP=A(II)
      IT=KEY(II)
      A(II)=A(IM)
      KEY(II)=KEY(IM)
      A(IM)=TEMP
      KEY(IM)=IT
      II=II-MM
      IF (II .GT. 0)  GO TO 11
    1 CONTINUE
      GO TO 20
  100 RETURN
      END
```

```
*****************************************************************************************************
*                                                                                                   *
*****************************************************************************************************

      SUBROUTINE FINDJ(A, NN, R, J)
C
C     << SUBROUTINE FINDJ FINDS THE INDEX OF ARRAY A  WHICH THE      >>
C     << VALUE OF IT IS CLOSEST TO R.                                >>
C
      REAL A(NN), R, VALUE
      INTEGER J

      VALUE = ABS(A(NN) - R)

      DO 100 I = NN-1, 1, -1

        IF (ABS(A(I) - R) .LT. VALUE) THEN
              VALUE = ABS(A(I) - R)
           ELSE
              J = I + 1
              RETURN
        ENDIF

  100 CONTINUE

      RETURN
      END
```

APPENDIX C. FORTRAN CODE FOR INTERVAL ESTIMATION PROCEDURE - WEIBULL CASE

```
      PROGRAM WEIBUL
C
C     ***************************************************************************
C     *                                                                         *
C     *    THIS PROGRAM COMPUTES THE RELIABILITY OF A SERIES SYSTEM             *
C     *    USING TWO DIFFERENT METHODS FOR THE CASE WHEN FAILURE                *
C     *    OF  EACH COMPONENTS IS WEIBULLY DISTRIBUTED.                         *
C     *    AND ALSO COMPARES THE ESTIMATE OF SHAPE PARAMETERS ( BETA )          *
C     *    WHICH IS COMPUTED FROM TWO DIFFERENT METHODS.                        *
C     *                                                                         *
C     *    THE SERIES SYSTEM CONSIDERED IN THIS PORGRAM HAS                     *
C     *    N COMPONENT TYPES AND EACH COMPONENT HAS SAME NUMBER OF              *
C     *    SAMPLE SIZES.                                                        *
C     *                                                                         *
C     *    BELOWS ARE GIVEN OR ASSUMED ;                                        *
C     *                                                                         *
C     *       BETA   ; SHAPE PARAMETER OF WEIBULL DISTRIBUTION                  *
C     *       ALPHA  ; SIGNIFICANCE LEVEL                                       *
C     *       X      ; TEST TIME                                                *
C     *       RSYS   ; RELIABILITY OF A SERIES SYSTEM                           *
C     *                                                                         *
C     *    THESE ARE VARIABLES USED ;                                           *
C     *                                                                         *
C     *       BETHAT ; BETA HAT FROM THE FORMULA                                *
C     *       BETNEW ; BETA HAT FROM NEWTON - RAPHSON METHOD                    *
C     *       CHISQR ; CHI-SQUARE VALUE FOR GIVEN ALPHA AND DF                  *
C     *       CLR    ; TRUE CONFIDENCE LIMIT FROM THE FORMULA                   *
C     *       CLRNEW ; TRUE CONFIDENCE LIMIT FROM THE N-R METHOD                *
C     *       EXPO   ; EXPONENTIAL RANDOM NUMBERS WITH MU=1                     *
C     *       KSUBM  ; K SUB M VALUE                                            *
C     *       LAMBDA ; SCALE PARAMETER OF WEIBULL DISTRIBUTION                  *
C     *       LAMBIG ; MAXIMUM VALUE OF LAMBDA FROM THE N-R METHOD              *
C     *       LAMHAT ; ESTIMATES OF LAMBDA FROM THE FORMULA                     *
C     *       LAMMAX ; MAXIMUM VALUE OF LAMBDA FROM THE FORMULA                 *
C     *       LAMMU  ; FINALLY COMPUTED LAMBDA FROM THE FORMULA                 *
C     *       LAMSTR ; LAMBDA TO THE BETA                                       *
C     *       LAMNEW ; ESTIMATES OF LAMBDA FROM THE N-R METHOD                  *
C     *       LAMUNR ; FINALLY COMPUTED LAMBDA FROM THE N-R METHOD              *
C     *       RATIO  ; RATIO OF LAMBDA FOR THE FORMULA                          *
C     *       RATNEW ; RATIO OF LAMBDA FOR THE N-R METHOD                       *
C     *       RHAT   ; R( 500 * (1-ALPHA) ) FOR THE FORMULA                     *
C     *       RHTNEW ; R( 500 * (1-ALPHA) ) FOR THE N-R METHOD                  *
C     *       XBENEW ; X TO THE BETNEW                                          *
C     *       XHAT   ; SUM OF XIJHAT                                            *
C     *       XIJHAT ; W TO THE BETHAT                                          *
C     *       XIJNEW ; W TO THE BETNEW                                          *
C     *       XNEW   ; SUM OF XIJNEW                                            *
C     *       XTOBET ; X TO THE BETHAT                                          *
C     *       W,WEIB ; WEIBULL RANDOM NUMBERS                                   *
```

```
C     *       ZALPHA ; RIGHT PERCENTILE POINT (NORMAL DISTRIBUTION)    *
C     *                                                                *
C     ******************************************************************
C
C
C     PARAMETER ( N=5, M=30 , NN= 500 )
      PARAMETER ( N=15, M=30 , NN= 500)

      REAL      BETA(N), LAMBDA(N), X(N), XHAT(N), XTOBET(N), XX(N,M)
      REAL      W(N,M), Y(N,M), BETHAT(N), WBHAT(N,M), LAMHAT(N)
      REAL      RATIO(N), LAMSTR(N), LAMMAX, RSYS(2), KSUBM(N), LAMMU
      REAL      EXPO(M), ALPHA(2), ZALPHA(2), RHAT(10), RSL(NN), CHISQR
      REAL      B(M), KEY(M), KEY1(NN), KEY2(NN), XIJHAT(N,M)
      REAL      WEIB(M), BETNEW(N), XBENEW(N), XNEW(N), XIJNEW(N,M)
      REAL      LAMNEW(N), LAMBIG, RATNEW(N), LAMUNR, RSLNEW(NN)
      REAL      RHTNEW(10), CLR(10), CLRNEW(10)
      REAL      BHTBAR, BNRBAR, BHTMSE, BNRMSE

      INTEGER   S(N), DF, ICOUNT, SAMPLE(N), CASE

      DATA ISEED  / 1736 /
      DATA ALPHA  / .2, .05 /
      DATA RSYS   / .90, .975 /

C     DATA BETA   / 1.2,1.4,1.6,1.8,1.5 /
C     DATA BETA /1.2,1.4,1.6,1.8,1.5,1.3,1.7,1.5,1.2,1.6,1.9,1.8,1.4,
C    *           1.6,1.1/
C     DATA BETA   / 2.2,2.4,2.6,2.8,2.5 /
      DATA BETA /2.2,2.4,2.6,2.8,2.5,2.3,2.7,2.5,2.2,2.6,2.9,2.8,2.4,
     *           2.6,2.1/

C     DATA X      / 2,3,7,8,10/
      DATA X /2,3,7,8,10,3,7,10,1,7,8,3,10,1,8/

C     DATA SAMPLE /9,7,10,8,6/
C     DATA SAMPLE /30,21,10,15,26/
C     DATA SAMPLE / 30,63,75,98,26 /

C     DATA SAMPLE /6,7,5,6,7,8,9,5,8,7,9,10,7,9,6/
C     DATA SAMPLE /3,7,10,15,20,15,7,5,30,20,10,7,13,21,30/
C     DATA SAMPLE /15,40,35,17,26,67,50,65,80,32,95,100,15,45,30/

      DATA SAMPLE / 15 * 15 /

      CASE = 1

      DO 10 I = 1, N
           IF ( SAMPLE(I) .LE. 15)  KSUBM(I) = 1.40
           IF ( SAMPLE(I) .LE. 30)  KSUBM(I) = 1.50
   10 CONTINUE


      DO 220 JJ = 1, 2

C       /* FINDING THE LAMBDA  AND LAMBDA STAR FROM THE GIVEN DATA    */
        DO 20 I = 1, N
```

```
            LAMBDA(I) = (( - ALOG(RSYS(JJ)) / N) ** (1./BETA(I)))/ X(I)
            LAMSTR(I) = LAMBDA(I) ** BETA(I)
            S(I)      = 0.84 * SAMPLE(I) - 0.5
   20    CONTINUE

C        /* SUBROUTINE ZTABLE COMPUTES THE RIGHT PERCENT POINT ZALPHA  */
C        /* FROM RIGHT CUMULATIVE PROBABILITY ALPHA                    */

         DO 40 I = 1, 2
            CALL ZTABLE(ALPHA(I), ZALPHA(I))
   40    CONTINUE


C        /* COMPUTE THE DEGREE OF FREEDOM FOR CHI-SQUARE               */

         DF = 0
         DO 60 I = 1, N
            DF = DF + SAMPLE(I)
   60    CONTINUE
         DF = 2 * DF


C        /* COMPUTE CHI-SQUARE VALUE FOR GIVEN ALPHA AND DF            */

         DO 200 LL = 1, 2

            IF ( DF .EQ. 1 ) THEN
                 P = ALPHA(LL) / 2
                 CALL ZTABLE(P, ZALPHA(LL))
                 CHISQR = ZALPHA(LL) ** 2

            ELSE IF ( DF .EQ. 2 ) THEN
                 CHISQR = -2 * ALOG(ALPHA(LL))

            ELSE IF ( DF .GE. 3 ) THEN
                 CHISQR = DF * (1 - 2./(9 * DF) +
     *                     ZALPHA(LL) * SQRT( 2./(9 * DF))) ** 3
            ENDIF

            LAMMAX = 0
            LAMBIG = 0

            DO 180 ITER = 1, NN

                 DO 140 I = 1, N

                      /* GENERATE EXPONENTIAL RANDOM NUMBER WITH MU=1  */
                      CALL LEXPN(ISEED, EXPO, SAMPLE(I), 1, 0)

                      /* CONVERT EXPONENTIAL RANDOM NUMBER WITH MU=1   */
                      /* TO MU=LAMBDA AND GET WEIBULL RANDOM NUMBERS   */
                      DO 80 J = 1, SAMPLE(I)
                           XX(I,J) = EXPO(J) / LAMSTR(I)
                           W(I,J)  = XX(I,J) ** (1/BETA(I))
                           Y(I,J)  = ALOG(W(I,J))
                           WEIB(J) = W(I,J)
```

```
                    B(J)    = Y(I,J)
 80               CONTINUE

                  /* SUBROUTINE SHSORT WILL SORT ARRAY B IN        */
                  /* ASCENDING ORDER BY SHELL SORT ALGORITHM       */
                  CALL SHSORT(B, KEY, SAMPLE(I))

                  /* FINDING BETA HAT (BETHAT) BY THE FORMULA      */

                  SUM1 = 0
                  SUM2 = 0

                  DO 100 J = 1, S(I)
                       SUM1 = SUM1 + B(J)
100               CONTINUE

                  DO 120 J = S(I)+1, SAMPLE(I)
                       SUM2 = SUM2 + B(J)
120               CONTINUE

                  BETHAT(I) = (SAMPLE(I)) * KSUBM(I) /
    *             (S(I)*SUM2 / (SAMPLE(I) - S(I)) - SUM1)

                  TEMP = BETA(I)

                  /* SUBROUTINE NEWTON WILL COMPUTE THE ESTIMATE   */
                  /* OF BETA (BETNEW) BY NEWTON-RAPHSON METHOD     */
                  CALL NEWTON(WEIB, SAMPLE(I), TEMP, BETNEW(I))

                  XTOBET(I) = X(I) ** BETHAT(I)
                  XBENEW(I) = X(I) ** BETNEW(I)

                  XHAT(I) = 0.
                  XNEW(I) = 0.

                  DO 130 J = 1, SAMPLE(I)
                       XIJHAT(I,J) = W(I,J) ** BETHAT(I)
                       XHAT(I)     = XHAT(I) + XIJHAT(I,J)

                       XIJNEW(I,J) = W(I,J) ** BETNEW(I)
                       XNEW(I)     = XNEW(I) + XIJNEW(I,J)
130               CONTINUE

                  LAMHAT(I) = SAMPLE(I) / XHAT(I)
                  LAMNEW(I) = SAMPLE(I) / XNEW(I)

                  IF (LAMHAT(I) .GT. LAMMAX ) THEN
                       LAMMAX = LAMHAT(I)
                  ENDIF

                  IF (LAMNEW(I) .GT. LAMBIG ) THEN
                       LAMBIG = LAMNEW(I)
                  ENDIF
140          CONTINUE

             SUM3  = 0
```

```
                SUM3N = 0
                SUM4  = 0
                SUM4N = 0

                DO 160 I = 1, N
                     RATIO(I)  = LAMHAT(I) / LAMMAX
                     RATNEW(I) = LAMNEW(I) / LAMBIG
                     SUM3  = SUM3 + RATIO(I) * XHAT(I)
                     SUM4  = SUM4 + RATIO(I) * XTOBET(I)

                     SUM3N = SUM3N + RATNEW(I) * XNEW(I)
                     SUM4N = SUM4N + RATNEW(I) * XBENEW(I)

160             CONTINUE

                LAMMU      = CHISQR / ( 2 * SUM3 )
                RSL(ITER) = EXP( - LAMMU * SUM4 )

                LAMUNR     = CHISQR / ( 2 * SUM3N )
                RSLNEW(ITER) = EXP( - LAMUNR * SUM4N )

180         CONTINUE

            CALL SHSORT(RSL, KEY1, NN)
            CALL SHSORT(RSLNEW, KEY2, NN)

            KK = ( 1 - ALPHA(LL)) * NN

            /* RHAT IS THE RELIABILITY OF THE SERIES SYSTEM COMPUTED   */
            /* BY THE FORMULA                                          */
            /* RHTNEW IS THE RELIABILITY OF THE SERIES SYSTEM COMPUTED */
            /* BY THE NEWTON - RAPHSON METHOD                          */

            RHAT(CASE)   = RSL(KK)
            RHTNEW(CASE) = RSLNEW(KK)

            /* SUBROUTINE FINDJ WILL FIND THE TRUE CONFIDENCE LIMIT    */
            CALL FINDJ(RSL, NN, RSYS(JJ), J1)
            CALL FINDJ(RSLNEW, NN, RSYS(JJ), J2)

            CLR(CASE)    = J1 / FLOAT(NN)
            CLRNEW(CASE) = J2 / FLOAT(NN)

            CASE = CASE + 1

200     CONTINUE


220 CONTINUE

    DO 240 I = 1, N
         WRITE(6,500) BETA(I), SAMPLE(I)
240 CONTINUE

    WRITE(6,550) N, NN
    WRITE(6,600)
```

```
      WRITE(6,650)
      WRITE(6,670)

      CASE = 1
      DO 280 JJ = 1, 2
           DO 260 LL = 1, 2

              WRITE(6,700) N,RSYS(JJ),ALPHA(LL),RHAT(CASE),RHTNEW(CASE),
     *                     CLR(CASE), CLRNEW(CASE)
              CASE = CASE + 1

  260      CONTINUE
  280 CONTINUE



  500 FORMAT(T10,'INITIAL BETA           :',3X,F4.1,
     *       T45,'NUMBER OF SAMPLE SIZES :',2X,I3)
  550 FORMAT(T10,'NUMBER OF COMPONENTS   :',2X,I3,
     *     /,T10,'NUMBER OF REPLICATIONS :',2X,I3)
  600 FORMAT('1',5(/),7X,'**** RELIABILITY OF SERIES SYSTEM ****')
  650 FORMAT(///,T35,'R1, CL1 ; USING THE FORMULAR',
     *         /,T35,'R2, CL2 ; USING THE NEWTON RAHPSON METHOD')
  670 FORMAT(///,T6,'NUMBER OF',T19,'RELIABILITY',T33,'ALPHA',T46,'R1',
     *       T56,'R2',T66,'TRUE',T74,'TRUE',/,
     *       T6,'COMPONENTS',T19,'OF SYSTEM',T66,'C.L. 1',T74,'C.L. 2',
     *       /,T5,76('-'))
  700 FORMAT(/,T8,I5,T22,F5.3,T33,F5.2,T43,F7.4,T53,F7.4,T65,
     *       F7.4,T73,F7.4)


      STOP
      END

*******************************************************************************
*                                                                             *
*******************************************************************************

      SUBROUTINE ZTABLE(ALPHA,ZALPHA)
C
C     << SUBROUTINE ZTABL2 COMPUTES RIGHT PERCENT POINT ZALPHA FROM  >>
C     << RIGHT CUMULATIVE PROBABILITY ALPHA                          >>
C
      REAL ALPHA, ZALPHA


      IF ((ALPHA .GT. 0.0) .OR. (ALPHA .LT. 1)) THEN

           W = - ALOG(4 * ALPHA * ( 1 - ALPHA ))
           ZALPHA = SQRT(W * (2.06118 - ( 5.72622 / (W + 11.6406))))

           IF (ALPHA .GT. 0.5) THEN
                ZALPHA = - ZALPHA
           ENDIF
```

```
      ENDIF

      RETURN
      END

*****************************************************************************************
*                                                                                       *
*****************************************************************************************

      SUBROUTINE SHSORT(A,KEY,N)
      DIMENSION A(N),KEY(N)
      M1=1
    6 M1=M1*2
      IF (M1 .LE. N)  GO TO 6
      M1=M1/2-1
      MM=MAX0(M1/2,1)
      GO TO 21
   20 MM=MM/2
      IF (MM .LE. 0)  GO TO 100
   21 K=N-MM
   22 DO 1 J=1,K
      II=J
   11 IM=II+MM
      IF (A(IM) .GE. A(II))  GO TO 1
      TEMP=A(II)
      IT=KEY(II)
      A(II)=A(IM)
      KEY(II)=KEY(IM)
      A(IM)=TEMP
      KEY(IM)=IT
      II=II-MM
      IF (II .GT. 0)  GO TO 11
    1 CONTINUE
      GO TO 20
  100 RETURN
      END

*****************************************************************************************
*                                                                                       *
*****************************************************************************************

      SUBROUTINE NEWTON(TIME, M, BETA, BETNEW )
C
C     << SUBROUTINE NEWTON WILL COMPUTE THE ESTIMATE OF BETA (BETHAT) >>
C     << BY NEWTON - RAPHSON METHOD                                   >>
C
      REAL      TIME(M), LNTIME(100), A(3), LNTS
      INTEGER   R

C     DATA TS / 6 /

      C    = 0
      ITER = 0
```

```
      R    = M
C     LNTS = ALOG(TS)

      DO 20 I = 1, R
           LNTIME(I) = ALOG(TIME(I))
           C = C + LNTIME(I)
   20 CONTINUE

      C = C / R

   30 DO 60 J = 1, 3
           SUM = 0

           DO 40 K = 1, R

                IF ( J .EQ. 1 ) THEN
                    SUM = SUM + TIME(K) ** BETA
                ELSE
                    SUM = SUM + ((TIME(K) ** BETA)*(LNTIME(K) ** (J-1)))
                ENDIF
   40      CONTINUE

           A(J) = SUM

   60 CONTINUE

      /* FUNCTION FPRIME IS THE DERIVATIVES OF FUNCTION F                */
      QUOT = A(2) / A(1)
      FPRIME = A(3) / A(1) - (QUOT**2) + ((1/BETA)**2)
      F = QUOT - (1 / BETA) - C

      /* BETA IS UPDATED EACH TIME AND CHECK IF IT CONVERGES             */
      BETA = BETA - F / FPRIME

      ITER = ITER + 1
      IF (BETA .GT. 25.) GOTO 100

      IF ( ABS(F) .GT. 0.0001 ) GOTO 30

      ALPHA  = (A(1) / R) ** (1/BETA)
      BETNEW = BETA

      RETURN

  100 WRITE(6,*)  'DID NOT CONVERGE'
      WRITE(6,*)  'TRY AGAIN WITH BETTER ESTIMATE OF BETA'

      RETURN
      END
```

```
      SUBROUTINE FINDJ(A, NN, R, J)
C
C     << SUBROUTINE FINDJ FINDS THE INDEX OF ARRAY A  WHICH THE      >>
C     << VALUE OF IT IS CLOSEST TO R.                                >>
C
      REAL A(NN), R, VALUE
      INTEGER J

      VALUE = ABS(A(NN) - R)

      DO 100 I = NN-1, 1, -1

        IF (ABS(A(I) - R) .LT. VALUE) THEN
              VALUE = ABS(A(I) - R)
           ELSE
              J = I + 1
              RETURN
        ENDIF

  100 CONTINUE

      RETURN
      END
```

APPENDIX D. FORTRAN CODE FOR INTERVAL ESTIMATION PROCEDURE - NORMAL CASE

```
      PROGRAM NORMAL
C
C     ***************************************************************************
C     *                                                                         *
C     *   THIS PROGRAM DETERMINE THE ACCURACY OF AN APPROXIMATE                 *
C     *   LOWER CONFIDENCE BOUND FOR  P( X > Y ).                               *
C     *                                                                         *
C     *   PROGRAM NORMAL IS THE CASE WHEN Y IS GIVEN A VALUE Y0                 *
C     *                  &  X IS NORMALLY DISTRIBUTED WITH                      *
C     *                  UNKNOWN PARAMETERS.                                    *
C     *                                                                         *
C     ***************************************************************************
C
C
      REAL Y0, P, ZP, MUX, SUMX, SUMX2, XBAR
      REAL TEMP1, TEMP2, TEMP3, SIGHAT
      REAL R(3), SIGMAX(3), ALPHA(2), X(100), Y(100)
      REAL RL1(54,1000), ARL1(1000), BRL1(54), KEY(1000), CIRL1(54)
      REAL T1RL1(54), T1CI1(54), T2RL1(54), T2CI1(54)
      REAL TRL1(2), TCI1(2)
      REAL ZHAT(1000), ZKNIFE(54), ZVAR(54), ZBAR(54)
      REAL MUJUNK(54)

      INTEGER NUMX(3), NUMY(3), CASE, LINE


      DATA  R /.95,.99,.995/,  SIGMAX /.5,1,20/,  NUMX /10,25,75/
      DATA  NUMY /10,25,75/ ,  ALPHA / .2, .1/
      DATA  ISEED, JSEED / 4875,7981 /
      DATA  Y0 / 400 /

      CASE = 0

C     /* II IS THE INDEX FOR R. R(1) = .95, R(2) = .99, R(3) = .995   */
      DO 500 II = 1, 3


C       /* P IS THE RIGTH(UPPER) CUMULATIVE PROBABILITY                */
        P = 1 - R(II)


C       /* SUBROUTINE ZTABL2 COMPUTES RIGHT PERCENT POINT ZP           */
C       /* FROM RIGHT CUMULATIVE PROBABILITY P                         */

        CALL ZTABL2( P, ZP )


C       /*  JJ IS THE INDEX FOR SIGMA OF X.                            */
C       /*  SIGMAX(1) = .5, SIGMAX(2) = 1, SIGMAX(3) = 3               */
```

```
      DO 400 JJ = 1, 3

        MUX = ZP * SIGMAX(JJ) + Y0

C       /*  KK IS THE INDEX OF NUMBER OF X.                          */
C       /*  NUMX(1) = 10, NUMX(2) = 25, NUMX(3) = 75                 */
        DO 300 KK = 1, 3


C         /*  LL IS THE INDEX FOR ALPHA.                             */
C         /*  ALPHA(1) = .2, ALPHA(2) = .1                           */
          DO 200 LL = 1, 2

            CALL ZTABL2(ALPHA(LL), ZALPHA)

            CASE  = CASE + 1
            TEMP  = (NUMX(KK) - 1.) / NUMX(KK)
            TEMP0 = SQRT(TEMP)
            MUJUNK(CASE) = MUX


C           /* REPLICATE 1000 TIMES FOR EACH CASES                   */
            DO 100 I = 1, 1000

C             /* USE NORMAL RANDOM NUMBER GENERATOR TO GET           */
C             /* NUMX(KK) NUMBER OF X.                               */

              CALL LNORM(ISEED,X,NUMX(KK),2,0)


C             /* THIS PART IS TO GET SAMPLE MEAN(XBAR) AND           */
C             /* SAMPLE VARIANCE(XVAR)                               */

              DO 50 MM = 1, NUMX(KK)
                X(MM)  = SIGMAX(JJ) * X(MM) + MUX
 50           CONTINUE


C             /* SUBROUTINE VAR WILL COMPUTE SAMPLE MEAN AND         */
C             /*                             SAMPLE VARIANCE         */

              CALL VAR(X, NUMX(KK), XBAR, XVAR)

C             /* NOW WE COMPUTE  THE LOWER CONFIDENCE BOUND           */

              TEMP1 = (XBAR - Y0) / SQRT(XVAR * TEMP0)
              TEMP2 = 1. / NUMX(KK) +
     *                ((XBAR - Y0) ** 2) / (2*(NUMX(KK)+1)* XVAR)
              TEMP1 = TEMP1 - ZALPHA * SQRT(TEMP2) * TEMP


C             /* SUBROUTINE ZTABL1 COMPUTE RIGHT CUMULATIVE PROBA-   */
C             /* BILITY FROM RIGHT PERCENT POINT.                    */

              CALL ZTABL1(TEMP1, ARL1(I))
```

```
                ARL1(I) = 1. - ARL1(I)

                CALL JKNIFE(X, NUMX(KK), Y0, ZHAT(I))

  100           CONTINUE

                CALL VAR(ZHAT, 1000, ZBAR(CASE), ZVAR(CASE))

                SIGHAT = 1./NUMX(KK) + ZBAR(CASE)**2 / (2*(NUMX(KK)+1))
                SIGHAT = SQRT(SIGHAT) * TEMP
                SIGHAT = ZBAR(CASE) / TEMP0 - SIGHAT * ZALPHA

                CALL ZTABL1(SIGHAT, ZKNIFE(CASE))

                ZKNIFE(CASE) = 1 - ZKNIFE(CASE)


C               /* NON-IMSL LIBRARY 'SHSORT' WILL SORT ARL1 BY SHELL    */
C               /* SORT ALGORITHM.                                      */

                CALL SHSORT(ARL1, KEY, 1000)

                DO 150 I = 1, 1000
                  RL1(CASE,I) = ARL1(I)
  150           CONTINUE


C               /* SUBROUTINE FINDJ FINDS THE INDEX OF ARL20 WHICH THE  */
C               /* VALUE OF IT IS CLOSEST TO R.                         */

                CALL FINDJ(ARL1, R(II), J)


C               /* DIVIDING THE INDEX BY 1000 WILL GIVE US TRUE         */
C               /* CONFIDENCE LEVEL.                                    */

                CIRL1(CASE) = J / 1000.

                MM = 1000 * (1 - ALPHA(LL))
                BRL1(CASE) = RL1(CASE,MM)

                CALL TNYVAL(R(II), SIGMAX(JJ), NUMX(KK),
     *               ALPHA(LL), ZALPHA, MUX, Y0, TRL1, TCI1)

                T1RL1(CASE) = TRL1(1)
                T1CI1(CASE) = TCI1(1)
                T2RL1(CASE) = TRL1(2)
                T2CI1(CASE) = TCI1(2)


  200         CONTINUE
  300       CONTINUE
  400     CONTINUE
  500 CONTINUE
```

```
      WRITE(6,600)
      WRITE(6,630)
      I = 1
      LINE = 0
      DO 550 II = 1, 3
        DO 550 JJ = 1, 3
          DO 550 KK = 1, 3

            IF (LINE .GE. 3) THEN
                WRITE(6,800)
                LINE = 0
            ENDIF

            WRITE(6,710) R(II), SIGMAX(JJ), NUMX(KK),
     *                   BRL1(I), BRL1(I+1), CIRL1(I), CIRL1(I+1),
     *                   ZKNIFE(I), ZKNIFE(I+1),
     *                   MUJUNK(I), MUJUNK(I+1)
            I = I + 2
            LINE = LINE + 1
  550 CONTINUE


      WRITE(6,610)
      WRITE(6,650)
      I = 1
      LINE = 0
      DO 560 II = 1, 3
        DO 560 JJ = 1, 3
          DO 560 KK = 1, 3

            IF (LINE .GE. 3) THEN
                WRITE(6,800)
                LINE = 0
            ENDIF

            WRITE(6,700) R(II), SIGMAX(JJ), NUMX(KK),
     *                   T1RL1(I), T1RL1(I+1), T1CI1(I), T1CI1(I+1)
            I = I + 2
            LINE = LINE + 1
  560 CONTINUE


      WRITE(6,620)
      WRITE(6,650)
      I = 1
      LINE = 0
      DO 570 II = 1, 3
        DO 570 JJ = 1, 3
          DO 570 KK = 1, 3

            IF (LINE .GE. 3) THEN
                WRITE(6,800)
                LINE = 0
            ENDIF
```

```
            WRITE(6,700) R(II), SIGMAX(JJ), NUMX(KK),
     *                   T2RL1(I), T2RL1(I+1), T2CI1(I), T2CI1(I+1)
            I = I + 2
            LINE = LINE + 1
  570 CONTINUE



  600 FORMAT('1',5(/),7X,'****  Y IS GIVEN A VALUE OF Y0  ****')

  610 FORMAT('1',5(/),7X,'****  Y IS GIVEN A VALUE OF Y0  ****',
     *       //,20X,'-- X IS TRUNCATED NORMAL WITH PROBABILITY .95 --')
  620 FORMAT('1',5(/),7X,'****  Y IS GIVEN A VALUE OF Y0  ****',
     *       //,20X,'-- X IS TRUNCATED NORMAL WITH PROBABILITY .90 --')

  630 FORMAT(3(/),5X,'   CASE    ',6X,'  RL1(1000*(1-ALPHA))    ',
     *     4X,'TRUE CONFIDENCE LEVEL',5X,'JACKNIFE',
     *     ///,5X,'R    SX   N',3X,'ALPHA = .2',12X,'.1',11X,'.2',
     *     12X,'.1',/,3X,13('-'),6X,25('-'),4X,21('-'),2X,16('-'),/)

  650 FORMAT(3(/),5X,'   CASE    ',6X,'  RL1(1000*(1-ALPHA))    ',
     *       4X,'TRUE CONFIDENCE LEVEL',///,5X,'R    SX   N',3X,
     *       'ALPHA = .2',12X,'.1',11X,'.2',12X,'.1',/,3X,13('-'),6X,
     *       25('-'),4X,21('-'),/)

  700 FORMAT(3X,F4.3,2X,F3.1,2X,I2,4(7X,F7.4))
  710 FORMAT(3X,F4.3,1X,F4.1,2X,I2,4(7X,F7.4),1X,2(2X,F6.4),2(2X,F9.5))
  800 FORMAT(/)

      RETURN
      END

*****************************************************************************************
*                                                                                       *
*****************************************************************************************

      SUBROUTINE ZTABL1(ZALPHA, ALPHA)
C
C     << SUBROUTINE ZTABL1 COMPUTE RIGHT CUMULATIVE PROBABILITY FROM  >>
C     << RIGHT PERCENT POINT.                                         >>
C
      REAL PI, ALPHA, ZALPHA

      PARAMETER ( PI = 3.141592 )


      IF ( ZALPHA .EQ. 0.0 ) THEN
           ALPHA = 0.5
      ELSE
           ALPHA = EXP( -2 * ZALPHA ** 2 / PI )
           ALPHA = SQRT(1 - ALPHA * (1 + 2*(PI-3)*ZALPHA**4/(3*PI**2)))
           ALPHA = 0.5 * ( 1 - ALPHA )

           IF ( ZALPHA .LT. 0.0 ) THEN
                ALPHA = 1 - ALPHA
```

```
            ENDIF
        ENDIF

        RETURN
        END

*****************************************************************************
*                                                                           *
*****************************************************************************

        SUBROUTINE ZTABL2(ALPHA,ZALPHA)
C
C       << SUBROUTINE ZTABL2 COMPUTES RIGHT PERCENT POINT ZALPHA FROM   >>
C       << RIGHT CUMULATIVE PROBABILITY ALPHA                           >>
C
        REAL ALPHA, ZALPHA

        IF ((ALPHA .GT. 0.0) .OR. (ALPHA .LT. 1)) THEN

            W = - ALOG(4 * ALPHA * ( 1 - ALPHA ))
            ZALPHA = SQRT(W * (2.06118 - ( 5.72622 / (W + 11.6406))))

            IF (ALPHA .GT. 0.5) THEN
                ZALPHA = - ZALPHA
            ENDIF
        ENDIF

        RETURN
        END

*****************************************************************************
*                                                                           *
*****************************************************************************
        SUBROUTINE TTABLE(ALPHA, NU, TALPHA)
C
C       << SUBROUTINE TTABLE COMPUTES RIGHT CUMULATIVE PROBABILITY      >>
C       << TALPHA FROM T-DISTRIBUTION BY GIVEN ALPHA AND NU             >>
C
        REAL ALPHA, ZALPHA, TALPHA, A1, A2, A3
        INTEGER NU

        IF ((ALPHA .GT. 0.0) .OR. (ALPHA .LT. 1)) THEN

            CALL ZTABL2(ALPHA, ZALPHA)

            A1 = (ZALPHA**2 + 1) / 4
            A2 = (5*ZALPHA**4 + 16*ZALPHA**2 + 3) / 96
            A3 = (3*ZALPHA**6 + 19*ZALPHA**4 + 17*ZALPHA**2 - 15) / 384

            TALPHA = ZALPHA * (1 + A1/NU + A2/NU**2 + A3/NU**3)

            IF ( ALPHA .GT. 0.5 ) THEN
                TALPHA = - TALPHA
            ENDIF
```

```
      ENDIF

      RETURN
      END

*****************************************************************************************************
*                                                                                                   *
*****************************************************************************************************

      SUBROUTINE SHSORT(A,KEY,N)
      DIMENSION A(N),KEY(N)
      M1=1
    6 M1=M1*2
      IF (M1 .LE. N)  GO TO 6
      M1=M1/2-1
      MM=MAXO(M1/2,1)
      GO TO 21
   20 MM=MM/2
      IF (MM .LE. 0)  GO TO 100
   21 K=N-MM
   22 DO 1 J=1,K
      II=J
   11 IM=II+MM
      IF (A(IM) .GE. A(II))  GO TO 1
      TEMP=A(II)
      IT=KEY(II)
      A(II)=A(IM)
      KEY(II)=KEY(IM)
      A(IM)=TEMP
      KEY(IM)=IT
      II=II-MM
      IF (II .GT. 0)  GO TO 11
    1 CONTINUE
      GO TO 20
  100 RETURN
      END

*****************************************************************************************************
*                                                                                                   *
*****************************************************************************************************

      SUBROUTINE VAR(Z, NUMZ, ZBAR, ZVAR)
C
C     << SUBROUTINE VAR WILL COMPUTE THE MEAN AND THE VARIANCE        >>
C     << OF ARRAY Z                                                   >>
C
      REAL Z(NUMZ), SUMZ, SUMZ2, ZBAR, ZVAR

      SUMZ  = 0
      SUMZ2 = 0

      DO 100 I = 1, NUMZ

         SUMZ  = SUMZ  + Z(I)
```

```
          SUMZ2 = SUMZ2 + Z(I) ** 2

  100 CONTINUE

      ZBAR = SUMZ  / NUMZ
      ZVAR = ABS( SUMZ2 / NUMZ - ZBAR ** 2 )

      RETURN
      END

*******************************************************************************
*                                                                             *
*******************************************************************************

      SUBROUTINE FINDJ(A, R, J)
C
C     << SUBROUTINE FINDJ FINDS THE INDEX OF ARRAY A  WHICH THE       >>
C     << VALUE OF IT IS CLOSEST TO R.                                 >>
C
      REAL A(1000), R, VALUE
      INTEGER J

      VALUE = ABS(A(1000) - R)

      DO 100 I = 999, 1, -1

        IF (ABS(A(I) - R) .LT. VALUE) THEN
              VALUE = ABS(A(I) - R)
           ELSE
              J = I + 1
              RETURN
        ENDIF

  100 CONTINUE

      RETURN
      END

*******************************************************************************
*                                                                             *
*******************************************************************************

      SUBROUTINE TNYVAL(R,SIGMAX,NUMX,ALPHA,ZALPHA,MUX,Y0,
     *                  TBRL1,TCIR1)
C
C     << SUBROUTINE TNYVAL IS THE CASE WHEN X IS TRUNCATED NORMAL     >>
C     << AND Y IS GIVEN A VALUE                                       >>
C

      REAL R, SIGMAX, ALPHA, TALPHA, MUX, P(2), ZP(2), A(2)
      REAL X(100), X1, TEMP1, TEMP2, Y0
      REAL TARL1(1000), KEY(1000), TRL1(1000), TBRL1(2), TCIR1(2)

      INTEGER NUMX, CASE, COUNT
```

```
      DATA P / .95, .90 /
      DATA ISEED, JSEED / 4875, 7981 /

      DO 400 I = 1, 2

         P(I) = 1 - P(I)
         COUNT = 0

         CALL ZTABL2(P(I), ZP(I))

         A(I) = MUX + ZP(I) * SIGMAX

         DO 200 J = 1, 1000

  100       CALL LNORM(ISEED, X1, 1, 2, 0)

            XTEMP = MUX + SIGMAX * X1

            IF (XTEMP .LE. A(I)) THEN
                COUNT    = COUNT + 1
                X(COUNT) = XTEMP
            ELSE
                GOTO 100
            ENDIF

            IF (COUNT .LT. NUMX) GOTO 100

            CALL VAR(X, NUMX, XBAR, XVAR)


C           /* NOW WE COMPUTE  THE LOWER CONFIDENCE BOUND                */

            TEMP1 = (XBAR - Y0) / SQRT(XVAR*(NUMX-1) / NUMX)
            TEMP2 = 1. / NUMX +
     *              ((XBAR - Y0) ** 2) / (2*(NUMX+1) * XVAR)
            TEMP1 = TEMP1 - ZALPHA * SQRT(TEMP2)


C           /* SUBROUTINE ZTABL1 COMPUTE RIGHT CUMULATIVE PROBA-         */
C           /* BILITY FROM RIGHT PERCENT POINT.                          */

            CALL ZTABL1(TEMP1, TARL1(J))

            TARL1(J) = 1. - TARL1(J)

  200    CONTINUE


C        /* NON-IMSL LIBRARY 'SHSORT' WILL SORT TARL1 BY SHELL           */
C        /* SORT ALGORITHM.                                              */

         CALL SHSORT(TARL1, KEY, 1000)

         DO 300 J = 1, 1000
           TRL1(J) = TARL1(J)
```

```
  300     CONTINUE

C         /* SUBROUTINE FINDJ FINDS THE INDEX OF TARL1 WHICH THE       */
C         /* VALUE OF IT IS CLOSEST TO R.                              */

          CALL FINDJ(TARL1, R, J)


C         /* DIVIDING THE INDEX BY 1000 WILL GIVE US TRUE              */
C         /* CONFIDENCE LEVEL.                                         */
          TCIR1(I) = J / 1000.

          MM = 1000 * (1 - ALPHA)
          TBRL1(I) = TRL1(MM)

  400 CONTINUE

      RETURN
      END


*****************************************************************************************
*                                                                                       *
*****************************************************************************************

      SUBROUTINE JKNIFE(X, N, Y0, ZHAT)
C
C     << THIS ROUTINE IS FOR THE 'JACKNIFE' METHOD.                              >>
C     << DUMMY PARAMETER IS ZHAT.                                                >>
C
      REAL    X(N),XOMIT(100),XOMBAR(100),XOMVAR(100),ZOMHAT(100),ZHAT
      INTEGER M, N

      M = N - 1
      SUM = 0.

      DO 200 I = 1, N
           DO 100 J = 1, M

                IF (J .GE. I) THEN
                     XOMIT(J) = X(J+1)
                ELSE
                     XOMIT(J) = X(J)
                ENDIF

  100       CONTINUE

            CALL VAR(XOMIT, M, XOMBAR(I), XOMVAR(I))

            ZOMHAT(I) = ( XOMBAR(I) - Y0 ) / SQRT(XOMVAR(I))
            SUM = SUM +ZOMHAT(I)

  200 CONTINUE

      ZHAT = SUM / N
```

```
RETURN
END
```

LIST OF REFERENCES

1. Harter, H. and Moore, A. H. (1968) *Maximum Likelihood Estimation, from Doubly Censored Samples, of the Parameters of the First Asymptotic Distribution of Extreme values,* Journal of the American Statistical Association, 63

2. Mann, N. R. (1968) *Point and Interval Estimation Procedure for the Two-Parameter Weibull and Extreme Value Distribution,* Technometrics 10

3. Lawless, J. F. (1978) *Confidence Interval Estimation for the Weibull and Extreme Value Distribution,* Technometrics 20

4. Johns, M. V. and Lieberman, G. J. (1966) *An Exact Asymptotically Efficient Confidence Bound for Reliability in the case of the Weibull Distribution,* Technometrics 8

5. Thoman, D. R., Bain, L. and Antle, C. (1970) *Maximum Likelihood Estimation, Exact Confidence Interval for Reliability and Tolerance Limits in the Weibull Distribution,* Technometrics 12

6. Schneider, H. and Weissfeld, L. (1989) *Interval Estimation Based on Censored Samples from the Weibull Distribution,* Journal of Quality Technology, 21(3)

7. Myhre, J., Rosenfeld, A. and Saunders, S. (1978) *Determining Confidence Bounds for Highly Reliable Coherent Systems Based on a Paucity of Component Failures,* Naval Research Logistics Quarterly, 25, June 1978

8. Lieberman, G. J. and Resnikoff, G. (1955) *Sampling Plan for Inspection by Variables,* Journal of American Statistical Association, 50

9. Folks, H. L.,Pierce, D. A. and Stewart, C. (1965) *Estimating the Fraction of Acceptable Product,* Technometrics 7

10. Barton, D. E. (1961) *Unbiased Estimation of a Set of Probabilities,* Biometrica 48

11. Owen, D. B. and Hua, A. (1977) *Tables of Confidence Limits on the Tail Area of the Normal Distribution*, Communications in Statistics B6

12. Church, J. D. and Harris, B. (1970) *The Estimation of Reliability Stress-Strength Relationships*, Technometrics 12

13. Downton, F. (1973) *The Estimation of $Pr(y < x)$ in the Normal Case*, Technometrics 15

14. Bain, L. and Engelhardt, M. (1987) *Introduction to Probability and Mathematical Statistics*, Duxbury Press

15. Quenouille, M. H. (1956) *Notes on Bias in Estimation*, Biometrika 43

16. Miller, R. G. (1974) *The Jacknife - a Review*, Biometrika 61

17. Engelhardt, M. and Bain, L, (1977b) *Simplified Statistical Procedures for the Weibull or Extreme-value Distribution*, Technometrics 19

18. Harris, B. and Soms, A. (1983) *A Note on a Difficulty Inherent in Estimating Reliability from Stress - Strength Relationships*, Naval Research Logistics Quarterly, Vol 30

19. Basu, A. (1964) *Estimates of Reliability for Some Distribution Useful in Life Testing*, Technometrics 6

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Professor W. Max. WOODS Naval Postgraduate School, Code 55Wo Monterey, CA 93943-5000	7
4.	Professor Harold J. Larson Naval Postgraduate School, Code 55La Monterey, CA 93943-5000	1
5.	CHOI, Moon-Soo SMC 1840 Naval Postgraduate School Monterey, CA 93943	1
6.	PARK, Rae-Yoon SMC 2329 Naval Postgraduate School Monterey, CA 93943	1
7.	LEE, Hyeon-Soo 411-225 DeungChon 2 Dong, KangSeo-Gu Seoul, KOREA 150-02	6
8.	PARK, Hun-Keun SMC 2997 Naval Postgraduate School Monterey, CA 93943	1
9.	KIM, Hyeong-Suk SMC 2075 Naval Postgraduate School Monterey, CA 93943	1
10.	KIM, Sook-Han SMC 2318 Naval Postgraduate School Monterey, CA 93943	1
11.	RYOO, Moo-Bong SMC 1845 Naval Postgraduate School Monterey, CA 93943	1

12. Library 2
Korea Military Academy
P.O.Box 77, Gongneung-Dong, Nowon-Gu,
Seoul, KOREA 139-799

13. Robert LIPPERT 1
P.O.Box 1461
Pebble Beach, CA 93953-1461

14. Naval Air Systems Command 2
Attn: R.C. Trakas (AIR-5165)
Department of the Navy
Washington, DC 20361-5160

15. CHOI, Seung-Soo 1
P.O.Box 35
Ceres, CA 95307